NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A FIRST-PRINCIPLES APPROACH TO MEASURING ROBUSTNESS IN COMMAND AND CONTROL SYSTEMS

by

David S. Cohick

March 2018

Thesis Advisor:	David L. Alderson
Second Reader:	Keenan D. Yoho

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2018	Master's Thesis 12-09-2016 to 03-31-2018

4. TITLE AND SUBTITLE

A FIRST-PRINCIPLES APPROACH TO MEASURING ROBUSTNESS IN COMMAND AND CONTROL SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S)

David S. Cohick

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

There is growing evidence that organizational architectures of military Command and Control (C2) systems are evolving from multiple layered, hierarchical (top-down) commands to more adaptable commands of networked teams. This research presents a "first principles" approach to developing a computational framework to measure and compare the organizational architecture of any military unit or commercial business. The developed construct takes the form of a game that imitates processes an organization must accomplish to reach an objective. Supported by mathematical analysis, we implement the framework as a simulation to measure the effectiveness of various organizational architectures. We explore performance advantages of different architectures when an organization's objective and/or operating environment change.

14. SUBJECT TERMS

organization theory, organizational architecture, quantitative framework, team of teams, organic architecture, robustness, command and control

15. NUMBER OF PAGES 97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

A FIRST-PRINCIPLES APPROACH TO MEASURING ROBUSTNESS IN COMMAND AND CONTROL SYSTEMS

David S. Cohick
Commander, United States Navy
B.S., United States Naval Academy, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2018

Approved by: David L. Alderson
Thesis Advisor

Keenan D. Yoho
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

There is growing evidence that organizational architectures of military Command and Control (C2) systems are evolving from multiple layered, hierarchical (top-down) commands to more adaptable commands of networked teams. This research presents a “first principles” approach to developing a computational framework to measure and compare the organizational architecture of any military unit or commercial business. The developed construct takes the form of a game that imitates processes an organization must accomplish to reach an objective. Supported by mathematical analysis, we implement the framework as a simulation to measure the effectiveness of various organizational architectures. We explore performance advantages of different architectures when an organization’s objective and/or operating environment change.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AQI	Al-Qaeda in Iraq
C2	Command and Control
COT	Computational Organizational Theory
CEO	Chief Executive Officer
DoD	Department of Defense
DYCORP	DYnamic Computational ORganizational Performance
JP-1	Joint Chiefs of Staff Publication 1
JSOTF	Joint Special Operations Task Force
NPS	Naval Postgraduate School
OODA	Observe, Orient, Decide, Act
OT	Organizational Theory

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

In this thesis, we consider the design of Command and Control (C2) architecture from the perspective of operations research. We investigate how one measures the performance of an organization in conducting a mission, and we contrast various C2 architectures in their ability to accomplish tasks. This research is inspired by retired Army General Stanley S. McChrystal's book *Team of Teams: New Rules of Engagement for a Complex World*. A key lesson in *Team of Teams* is that the C2 organization and methods that work well in one environment for a given mission might not work well in a different environment and/or for a different mission. The implication is that an organization that is not adapted to its environment is bound to perform poorly.

This thesis presents a first-principles approach to creating a quantitative framework, in the form of a game, which imitates the basic processes of an organization. Using a lexicon familiar to Organization Theory, we examine the processes and structures of organizations to identify how they affect the way an organization performs its mission, solves problems, and maximizes productivity and efficiency. Specifically, we focus on the number and types of people in an organization, their interactions, and the internal processes within an organization. We implement our game using a Monte-Carlo style simulation that mimics organizational processes and yet is simple enough to explain and be supported by mathematical analysis. Specifically, we consider the situation where an organization is tasked with matching the letters in a given word when sampling from a random distribution of letters. The game measures the expected number of discrete time steps it takes an organizational architecture to complete its objective in a particular environment.

We use the notation and mathematics of Discrete-Time Markov Chains to quantitatively measure the performance of an architecture. Using this framework we represent game play as a series of random transitions from one state to another. We begin by examining simple architectures and incrementally considering more complicated ones.

We start with a single agent, working alone to complete match a single letter objective. Using mathematics and simulation we measure this most simple architecture and establish our foundation for incremental expansion. Our research continues to explore multiple

independent players. These players attempt to match identical tasks in parallel without any interaction. We show that adding independent (non-connected) workers to an architecture improves overall performance, but at a much slower rate than if the workers were connected in a manner that allows them to communicate and cooperate. We show that performance improves significantly when players can share letters between them.

Challenges arise as an architecture's size grows. As workers are added, there is an exponential growth in the potential number of connections between them; an organization that includes all such connections eventually ultimately suffers in performance. We show how hierarchies avoid this by carefully selecting connections between workers, thereby maximizing performance advantages of connections while minimizing costs. Because these larger organizational architectures and/or objectives quickly result in complicated mathematics, we rely more heavily on simulation as a means for analysis.

We examine hierarchies that have limited, one-way communication from the worker at the bottom to the manager at the top. We show why such hierarchies are commonly designed for organizations with specific, repetitive objectives and that operate in a stable environments. We study a series of architectures and demonstrate the performance impact of choosing the correct set of tasks to assign workers.

Next, we consider architectures with two-way communication rules for sharing letters with other workers via a manager. Now, unused letters are sent to the manager and distributed to workers who need them most. Such cooperation yields increased performance and less sensitivity to the set of tasks that are assigned to workers.

We also explore architecture robustness, namely how architectures perform in the presence of changing objectives or changing environments. To do so we introduce a two-tier hierarchy with a more complex objective. We use our simulation to measure its performance and then repeat measurements through a series of experiments in which we add more workers, vary worker task assignments, perturb letter distributions, and change objective words. This first set of experiments investigates the effects of changing task assignments. We find that when an objective and letter distribution are known, a specific set of task assignments can be chosen to achieve best performance. Second, we change the operating environment by switching letter distributions. We find that highly customized architectures perform well when operating in a stable environment, but for an architecture to be robust, it must equally assign tasks to account for any possible letter distribution. Lastly, we change an organization's objective and test its performance. In some cases, we find that customized architectures can perform very poorly or even fail to complete their objective. To increase robustness, we introduce the concept of a worker without an assigned task who can match any letter not already assigned.

Through the lens of operations research, we understand why the Joint Special Operations Task Force (JSOTF) originally failed in Iraq and why General McChrystal's "Team of Teams" concept was a better fit. We assert that organizations with a customized architecture tailored to complete a specific objective in a stable environment will perform most efficiently, however performance will suffer in the case of volatility in the objective or environment. Like a customized hierarchy tailored toward a specific objective in a stable environment, the JSOTF's performance against Al-Qaeda in Iraq was originally unsuccessful because its architecture was too specialized and less robust to external changes.

As General McChrystal describes in *Team of Teams*, his restructuring efforts changed the JSOTF's organizational architecture. The architecture became decentralized with decision making authorized at lower levels, thereby freeing up available work capacity for senior leaders. Additionally, the task force's architecture maximized the advantages of cooperation while limiting connections between workers by using teams and connecting them horizontally. McChrystal's new task force saw increased effectiveness fighting a fluid enemy in multiple arenas. Its architecture was defined by a centralized structure with a strict hierarchy of authority. The task force was comprised of specialized units that didn't have the connections and processes to effectively share information or cooperate due to the vertical nature of its architecture. Through our research we address why, while being extremely efficient at their assigned objectives, organizational architectures like that of the JSOTF, are less effective against an unpredictable and agile enemy.

Lastly, motivated by McChrystal's story, we comment on features of resilient architectures to understand the organizational changes described in *Team of Teams*. Perhaps the most important piece of the *Team of Teams* narrative is not the final topology of the task force's architecture, but the importance of an organization's ability to identify the need for change, know what protocols are necessary to effect the needed change, and initiate the change in time for it to be relevant.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

My chief acknowledgment is deservedly for my wife, Julia. Thank you for your steadfast love and support! Without your sensible understanding and numerous sacrifices during this year long effort I could not have completed this thesis.

Second, I thank Professor David Alderson for embracing my ambition to tackle an unconventional operations research topic. Your guidance, thoughtful contributions, and concrete support throughout the past year were crucial. Thank you for pushing me hard and helping me to explore organization theory and design from a fresh, analytic perspective. The process of developing a broad idea into a manageable and academically worthy topic was a lengthy but valuable experience. You have taught me that forging new steel is often harder than reshaping existing metal, but the lessons learned along the challenging journey can be more valuable than the end product.

I also sincerely thank Professor Keenan Yoho (Rollins College), Dr. Imre Balogh, and Professor Thomas Seager (Arizona State University) for their insightful comments and enthusiastic encouragement throughout this endeavor.

Last but not least, I thank Adam Hepworth, Chuck Clark, Ezra Akin, Lee Huffstetler, and Clark Petri for your friendship, support, and humor. Together we encouraged and supported each other and I'm so grateful to have shared this "academic fox-hole" with you five!

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

> *"To win we had to change. Surprisingly, that change was less about tactics or new technology than it was about the internal architecture and culture of our force."*
>
> — General S. McChrystal, Team of Teams (2015)

1.1 Motivation

In this thesis, we consider the design of Command and Control (C2) architecture from the perspective of operations research. We investigate how one measures the performance of an organization in conducting a mission, and we contrast various C2 architectures in their ability to accomplish complicated tasks.

This research is inspired by retired Army General Stanley S. McChrystal's book titled, *Team of Teams: New Rules of Engagement for a Complex World* (McChrystal et al., 2015). On July 15, 2015, McChrystal spoke to the student body of the United States Naval Postgraduate School (NPS), as part of the Secretary of the Navy Guest Lecture Series, about the story told in *Team of Teams*. In his lecture, McChrystal explains the challenges he faced after taking command of the Joint Special Operations Task Force (JSOTF) in Iraq beginning in September 2003. Despite being superiorly trained and equipped, his forces were unsuccessful in combating Al-Qaeda in Iraq (AQI), a very aggressive and unpredictable foe. The book details the transition of the JSOTF from a traditional military organization, reliant on planning and rehearsal, into a flatter, more integrated, and team oriented architecture that was able to aggregate information and adapt to external events faster than the enemy.

The story told in *Team of Teams* is relevant to both military and civilian organizations. Military units and civilian corporations alike are comprised of people, internal processes, and reporting relationships that were designed, or have evolved, to accomplish a mission.

A key lesson in *Team of Teams* is that the C2 organization and methods that work well in one environment for a given mission might not work well in a different environment and/or for a different mission. The implication is that an organization that is not adapted to its environment is bound to perform poorly. The story of the JSOTF's organizational transformation to overcome the challenges posed by AQI is presented as an exemplar for how modern organizations ought to adapt in the face of growing operational complexity.

Using Organization Theory (OT) we examine the processes and structures of organizations to identify how they affect the way an organization performs its mission, solves problems, and maximizes productivity and efficiency. Specifically, we focus on the number and types of people, their interactions, and the internal processes within an organization. In OT, this is referred to as *Organizational Architecture* or *Organizational Structure*. In this thesis, we use the term *architecture* because it implies more than just the structure of how people are organized (as seen on an organizational chart), it includes all communication connections, authority relationships, and internal processes.

In response to the JSOTF's lack of effectiveness combating AQI, McChrystal implemented major changes to its architecture and culture to increase adaptability in its new complex and rapidly changing environment. Large civilian organizations alike are changing their organizational architectures in order to remain relevant and profitable. But how does one know if an organization is appropriately adapted for a given mission and/or environment? Moreover, is it possible to design an optimal C2 architecture? And how can the field of operations research help to understand these issues?

1.2 Thesis Objectives

In this research, we investigate how to measure an organization's effectiveness in conducting a mission. We examine how changing an organization's mission and/or operating environment impacts performance. To do so, we introduce a simplified and stylized game as a proxy for the tasks and environment faced by an organization. We study various C2 architectures in terms of their ability to play this game. We analyze their performance both analytically and via simulation.

In this thesis, we re-tell McChrystal's story from an operations research perspective. We comment on features of an agile architecture that allow an organization to remain effective across a complex and fluid set of objectives and environments.

The remainder of this thesis is organized as follows. In Chapter 2, we review relevant literature in organizational theory, military C2, network warfare, and network science. In Chapter 3, we introduce our game, its design, and how we measure the performance of a given C2 architecture when playing it. In Chapter 4, we present our results supported by analytic and numerical experiments. In Chapter 5, we summarize our findings and discuss opportunities for additional research.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Literature Review

2.1 Military Command and Control

Command and Control (C2) means different things to different people. The complexities of its processes and systems form the underlying fabric that allow military units, large and small, to function in a coordinated fashion. As our adversaries adopt networked C2 architectures, so must those who seek to destroy or disrupt them. To do so, we require an understanding of how organizations are structured and how vital C2 architecture is to success against a dynamic enemy.

According to the U.S. Joint Chiefs of Staff Publication 1 (JP-1), to *command* is defined as "The exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission" (United States Joint Chiefs of Staff, 2017). "*Command and control* is the means by which a commander synchronizes and/or integrates force activities in order to achieve unity of command" (United States Joint Chiefs of Staff, 2017). JP-1 continues to describe C2 accordingly:

> C2 ties together all the operational functions and tasks and applies to all levels of war and echelons of command. C2 functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission.

In their NPS thesis titled, "Command and Control: An Introduction," Bethmann and Malloy (1989) discuss the complex nature and historical significance of C2. In their lexicon, effective C2 is "the net result of the successful interaction of a complex architecture that is comprised of people, procedures, and equipment" (Bethmann and Malloy, 1989). They contrast effective C2 with unfortunate examples of breakdowns in military C2. For example, the 1983 invasion of Grenada (Operation URGENT FURY) showcased communication and coordination difficulties between the Air Force, Army, Navy, and Marine Corps. During the initial days of the operation, the four services used different radio frequencies for controlling military strikes. The initial C2 architecture in place required Army ground units to contact headquarters in North Carolina and then messages were relayed via satellite to Navy leadership aboard aircraft carriers (Bethmann and Malloy, 1989). As a result of these C2 failures and similar challenges during the failed 1980 Iranian hostage rescue mission, the 1986 Goldwater-Nichols Act was passed to reorganize the entire military's chain of command. These sweeping changes marked the largest change in military C2 since the services were first created (Lederman, 1999).

Painter et al. (2009), in their NPS thesis titled, "Reorganizing for Irregular Warfare," present five domains to frame their discussion of C2 and organizational design: (1) Structure, (2) Environment, (3) Work Processes, (4) Human Resources, and (5) Culture. Of these, we focus attention on the following.

Structure. Painter et al. (2009) defines structure as "How an organization is designed to facilitate information flow and complete its work (tasks)." An organization's structure is commonly represented on a chart that uses boxes and lines to group organization members into assigned roles, depict formal reporting relationships, and state spans of control (i.e., the number of employees reporting to a supervisor; see Daft, 2001). Additionally, organizational structure divides "labor into distinct tasks and then achieves coordination among them" (Mintzberg, 1989). Example structures include mechanistic, organic, matrix, virtual, clusters, network, funtionalized, and divisionalized (Morgan, 2006).

Environment . All organizations operate in an environment. "Organizational environment is defined as everything that exists outside the boundary of the organization and has the potential to affect all or part of the organization" (Daft, 2001). The environment includes elements such as trends in industry, government restrictions, customer pressures, quantity and quality of resources and/or information, the financial community, etc. Morgan (2006) incorporates an important temporal element in her definition; the "degree of stability or change, abundance or scarcity of key resources, competition, political/legal/technological/-social/market conditions."

Work Processes. Work processes are the means by which an organization transforms raw material, information, and/or other inputs into some desired output. For example, a furniture maker uses fabric, wood, and other material to assemble an end product like a sofa. This work process is called *manufacturing*. To better understand *work processes*, consider the two following scenarios in which a Sailor services multiple helicopter engines: 1) The Sailor performs the first step of the servicing procedure sequentially on each engine before starting step 2. He performs one step on each engine on each rotation. 2) The Sailor performs all the servicing steps on the first engine before moving on to the next one. He completes one engine at a time until all are serviced. The Sailor in both scenarios is assigned the same task (service the helicopter engines) but accomplishes it using two different work processes.

When an organization is divided into different departments or divisions, work processes establish the level of interaction between groups. If a single department can accomplish its task without "outside" assistance then very little interaction with other departments is needed. However, if a task requires collaboration between multiple sub-organizations, appropriate work processes must exist to enable them to work together. This type of work process "requires extensive horizontal linkages in the form of liaisons to coordinate activities" (Painter et al., 2009).

As noted by Galbraith et al. (2001), "Organizational architecture is about the structure, processes, organizational roles, power and authority, reporting relationships... not people practices (staffing and selection, performance feedback, learning and development) or strategy (vision, direction, competitive advantage) or reward systems (goals, scorecards, and metrics... values and behaviors, and compensation/rewards)." For this reason, we focus on *organizational architecture* because it represents tangible components of organizational design that can be concretely modeled and manipulated.

2.2 Three Dimensions of C2 Architecture

Alberts and Hayes (2006) identify the following three dimensions of C2 architecture.

2.2.1 Allocation of Decision Rights

The key question here is: *Who gets to make what type of decisions?*

Members of an organization who have different jobs or perform specific roles will have varying levels of authority to make decisions commensurate with their position. Decision making authority of a more *centralized* organization resides at higher levels (e.g., a CEO or

Commanding Officer). In a *decentralized* organization, the authority to make decisions is disseminated lower among the ranks.

The allocation of decision rights is usually a function of rank, position, experience, or established rules and regulations. Decision rights also are impacted by time and circumstances. In the military, a person's decision making authority is commonly associated with rank or position, but this is not always the case. Often, a lower-ranking soldier must make a potentially risky or costly decision because he is authorized to do so when certain circumstances are met and there is not adequate time to seek approval from superiors.

2.2.2 Distribution of Information

The key question here is: *Who knows what? Who gets which resources?*

Alberts and Hayes (2006) use the term *information* to include knowledge, data, experience, and understanding. We include an organization's physical resources in this discussion as well, because the proper allocation of resources to the correct person(s) can directly impact an organization's effectiveness. Typically, the amount and type of information or resources that a member of an organization is allocated varies upon the task or role assigned.

The distribution of information and resources is linked with the allocation of decision rights. An organization with more decentralized decision making requires increased information sharing/access so those empowered to make decisions can more effectively decide correctly. In more hierarchical organizations, only top leaders need access to all the information since they are responsible for most decision making.

The spectrum of information distribution has three extremes: 1) One person has all the information and resources and allocates them as needed; 2) Everyone has access to all information and resources; or 3) People in different parts of an organization have information and resources that are not made available to others, resulting in no one having access to all of it. Depending on the organization's strategy and objective, its rules for allocation will fall somewhere in-between.

2.2.3 Patterns and Policies Governing Interactions

The key question here is: *Who communicates or interacts with whom? What rules are in place that constrain or enhance collaboration?*

The patterns and policies governing interactions within an organization are the foundational mechanisms that define an organization's hierarchical structure and degree of collaboration. In its simplest form, this idea can be visualized as the formal policies and reporting relationships shown as lines on an organizational chart. But not every interpersonal connection or informal collaboration can be formalized and depicted on a chart. These informal interactions and practices "can be considered the fuzzy connections you don't see between people in the 'white space' of an organizational chart" (Galbraith et al., 2001).

In more hierarchical organizations such as a large department store, the owner communicates with the general manager, who leads department managers. The department managers interact with their respective clerks to execute day-to-day business. In such an organization, the patterns and policies do not usually provide a means for a clerk to communicate with the owner without going through his or her chain of command. In less hierarchical organizations there are fewer layers of authority, resulting in fewer vertical (up and down layers) linkages but increased numbers of horizontal (laterally across a layer) connections. This results in increased communication and collaboration, essential for organizations that must change due to pressures received from their environments (French and Bell, 1999).

2.3 Background on Organization Theory

In parallel to the work on military organizations and C2, there is a large body of research on organizations. Daft (2001) defines *organizations* as "(1) social entities that (2) are goal oriented, (3) are designed as deliberately structured and coordinated activity systems, and (4) are linked to the external environment." The study of an organization's design, strategy, and leadership is known as Organization Theory (OT), which centers on "creating a community of collective effort that yields more than the sum of each individual's efforts and results" (Galbraith et al., 2001).

Organizational theory considers a wide variety of organizational architectures, each of which is "the complex product of [its] history, strategy, and environmental circumstances"

(Nadler et al., 1992). We now review historical perspectives of OT.

2.3.1 Scientific Management Perspective

Frederick W. Taylor's "Scientific Management" theory was the prominent architecture type during the early years of OT. Popularized in his 1911 book, *The Principles of Scientific Management*, Taylor's theory encourages "scientifically determined jobs and management practices as the way to improve efficiency and labor productivity" (Daft, 2012). This type of organizational architecture dominated OT during the Industrial Revolution because it focused on efficiency and productivity.

Often known as Taylor's "one best way" movement, the scientific management perspective postulates that all jobs and organizational decisions be designed to find the optimum way to complete a task (Taylor, 1911). His theory recommends "managers develop precise, standard procedures for doing each job, select workers with appropriate abilities, train workers in the standard procedures, carefully plan work, and provide wage incentives to increase output" (Daft, 2001).

The scientific management architecture is hierarchical with a leader at the top, one or more layers of management, and workers who occupy the lowest level, as seen in Figure 2.1. It is characterized by two or more layers whose members are connected vertically with those who are superior and/or inferior to them. It has very few (if any) lateral connections within a layer. This architecture is well-suited for organizations with objectives that are decomposable into repetitive tasks that do not require significant skill or problem solving for a worker to perform. For this reason, scientific management architectures are sometimes referred to "mechanistic" because of the machine-like efficiencies they strive to achieve.

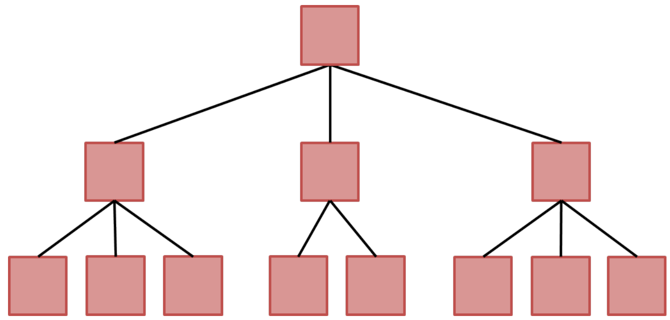


Figure 2.1. Example Hierarchy Architecture

An example organization that would benefit from a scientific management style architecture is a notional single product factory. This organization's objective is to manufacture one product as many times as possible with great efficiency in order to maximize profit. Factory workers perform their assigned task, following strict repetitive procedures while managers improve methods, tools, or procedures to further streamline worker processes. Complicated tasks are split up and assigned to multiple workers in order to increase efficiency.

Using Albert's three dimensions of organizational architecture, we characterize scientific management style architectures:

1. *Allocation of Decision Rights*:
   This type of architecture is mostly hierarchical with the majority of the decision making power and responsibility residing with senior leadership. Managers execute their role of continually striving for increased output by improving processes, procedures, and employee training. Managers funnel questions and problems to top leaders for resolution if a problem arises they don't have a prescribed solution for. Similarly, low-level employees receive specific tasks to which they are given just enough authority to accomplish. Workers follow detailed procedures from which they are not allowed to deviate.
2. *Distribution of Information*:
   Similar to decision rights, senior leadership controls the majority of information. In a hierarchy, connections between people are primarily vertical, spanning up or down a single layer at a time. This means that information received in one part of the organization must travel vertically as high as necessary to then trickle down to other

low-level employees “across” an organization. Low-level employees require no more information or resources necessary beyond what’s needed to perform their tasks.

3. *Patterns and Policies Governing Interactions*:
   Hierarchical architectures are very effective at accomplishing a single objective at a large scale because superfluous interactions between members are avoided. By nature of Taylor’s system, the manager’s primary function is to constantly seek opportunities for improving efficiency. As a result, any function or characteristic that differentiates a worker from a machine is minimized (Mintzberg, 1989).

The benefits of scientific management are realized by organizations which have a singular (or very small set of) objectives and operate in a stable environment. Manufacturers benefit from “Taylorism” because complicated objectives (i.e., building a car) are broken into smaller tasks which specialized workers can do quickly.

The downfalls of scientific management reside in its slowness to adapt to changes in the organization’s objective or environment. A factory designed to build military tanks uses specialized equipment and has workers trained to accomplish tank-building related tasks. If the factory is forced to begin building boats, the existing procedures and divisions of labor would no longer be effective. This should come as no surprise; workers are like “single-purpose mechanisms designed to transform specific inputs into specific outputs and can engage in different activities if they are explicitly modified or redesigned to do so” (Morgan, 2006). Organizations using the principles of scientific management have high productivity and efficiently use resources to maximize output but their specificity limits flexibility and innovation in dynamic environments.

2.3.2 Bureaucratic Organizations

There is another OT perspective focusing on efficiency that emerged during the Industrial Revolution. Bureaucracy Theory, popularized in Max Weber’s posthumously published book titled *Economy and Society* (Weber, 1922), emphasizes “designing and managing organizations on an impersonal, rational basis through such elements as clearly defined authority and responsibility, formal record-keeping, and uniform application of standard rules” (Daft, 2012). Bureaucracy theory combines Taylor’s scientific management with Henri Fayol’s proposed 14 administrative principles of management. Together, these perspectives "formed the foundation for modern management practice and organizational design" until the 1980s (Daft, 2012).

2.3.3 Hawthorne Studies

A series of experiments investigating human aspects of work and working conditions were conducted at the Hawthorne plant of the Western Electric Company between 1927 and 1932 (Carey, 1967). The work of Elton Mayo and other contributors suggest that social and psychological needs are important in motivating employees. This was a major departure from previous mechanistic theories and "the publication of these findings led to a revolution in worker treatment and laid the groundwork for subsequent work examining treatment of workers, leadership, motivation, and human resource management" (Daft, 2012).

Roethlisberger and Dickson (1939) interpret the Hawthorne Studies and realize intangible actions and behaviors are present in organizations. "Specifically, their observations about the presence of informal structure - unofficial relationships within the work group - constituted the simple realization that mutual adjustment served as an important coordination mechanism in all organizations" (Mintzberg, 1989). Their work highlighted the importance of including human relations and worker well-being in future organizational theories.

2.3.4 Contingency Theory

Scholars of contingency theory investigate relationships between an organization's architecture and its environment. "Contingency means that one thing depends on other things, and for organizations to be effective there must be a 'goodness of fit' between their structure and various contingency factors" (Daft, 2012). According to Daft (2012), such factors include: size, organizational technology, environment, goals and strategy, and culture.

In a popular 1967 study, Harvard researchers Paul Lawrence and Jay Lorsch examined multiple categories of American companies. As documented by Mintzberg (1989), Lawrence and Lorsch found that "environmental conditions surrounding the organization affected its choice of [architecture] significantly". For example, Lawrence and Lorsch determined shipping container companies are well-suited for stable operating environments and had mechanistic architectures whereas plastic industry businesses operate in more dynamic environments and require architectures which facilitate collaboration and flexibility (Mintzberg, 1989). In general, contingency theory centers around the concept that there is no one best architecture. What works for one organization may not work for another. "Contingency theory means 'it depends' " (Daft, 2001).

2.3.5 Organic Organizations

Beginning in the 1970s, the rate of technological change increased rapidly, causing mechanistic and bureaucratic organizations to struggle. A new theory of organizational architecture, coined "organic" by Tom Burns and G.M. Stalker in 1961, emerged which was human-centric and distinguished itself from previous OT perspectives by its "absence of standardization in the organization" (Mintzberg, 1989). Organic architectures rely on the human elements in an organization, not formal protocols and procedures. They are more adaptive and best suited for organizations needing rapid innovation and flexible responses to pressures from working in dynamic operating environments.

Organic architectures benefit from decentralization which empowers decision making at lower levels in an organization. Members are given roles not jobs, with responsibilities and not procedures. This less formal atmosphere encourages discussion and collaboration among peers. Organic organizations have architectures that emphasize horizontal communication and interactions to ensure information flows across all groups and teams. "The widespread sharing of information enables all employees to have complete information about the company so they can act quickly" (Daft, 2012).

2.3.6 The Rise of Teams

The discussion of team-based work processes in OT began with organic architectures. Teams are the "fundamental work unit" (Daft, 2012) of organic organizations because they encourage collaboration and innovation. "Teams bring people together to work interdependently and share collective responsibility for outcomes" (Galbraith et al., 2001). Figure 2.2 is a graphical representation of a possible team architecture.

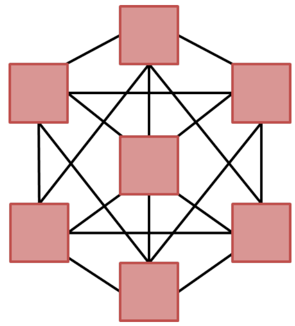


Figure 2.2. Example Team Architecture

Galbraith et al. (2001) distinguishes teams into three categories:

Issue Teams. These teams are assembled temporarily to solve a specific problem or accomplish a time-critical task. When complete, the team is disbanded and members resume their original roles in the organization.

Work Groups. These teams are clusters of people in an organization who share a common goal or do highly-related work. Being assigned to the same team enhances collaboration and accountability.

Cross-Business Teams. Cross-business teams have representation from multiple groups, bringing different perspectives from across an organization into the team. These teams support rapid innovation because of the diverse set of knowledge different members contribute. "[Cross-business teams] are an essential component of an integrated, flattened organization" (Galbraith et al., 2001).

Using Albert's dimensions of C2 architecture, we characterize team-based organizations:

1. *Allocation of Decision Rights*:
   In an organic architecture decision making is decentralized and often delegated to lower organizational levels. "People are encouraged to take care of problems by working with one another. . . using their discretion to make decisions" (Daft, 2012). Teams are granted necessary freedoms to seek solutions to their problems without having to get approval from superiors.

2. *Distribution of Information*:
   Information in organic organizations is shared quickly and made available to anyone who may benefit. Organic architectures emphasize "horizontal communication, with information flowing in all directions within and across departments and hierarchical levels" (Daft, 2012). In addition to sharing "bottom-up" information, strategy and guidance from superiors are disseminated across the entire organization to empower all members to make decisions that best support the organization. In the military we call this "knowing Commander's Intent."
3. *Patterns and Policies Governing Interactions*:
   Organic architectures have few formal rules which might constrain member actions and innovation. The absence of formalization allows an organic architecture to self-organize and adapt to accomplish the task at hand. Instead of departments of like-minded workers, organic architectures utilize teams comprised of people with different skills and knowledge to address a common goal.

Organic, team-based architectures allow an organization to adapt to changes. They are best suited for smaller organizations that rely on innovation and creating new solutions to problems. Example organic organizations might include small entrepreneurial firms and technology start-ups. The larger an organization grows, the more oversight and supporting roles (human resources, information technology, accounting, etc.) are needed. These larger companies, like Google and Facebook, cannot operate truly organically but they incorporate teams, information sharing, and empowerment to their advantage.

2.4 Our Approach and Other Considerations

Organization theory scholars and researchers strive to better model and understand organizational design, strategy, and leadership. The majority of work in this field is primarily qualitative because the nature of the concepts lend themselves poorly to quantitative discussions. This research establishes a foundation for such quantitative discussions in the form of a framework for measuring an organization's architecture's goodness-of-fit with its environment. To do this, we develop a framework in the form of a simple game that imitates the basic processes of an organization (gather, filter, assemble, deliver). Operations research tools allow us to explore OT from a new perspective and provide credibility to our simulation findings. Our effort to design a quantitative measure of organizational architecture goodness-of-fit is a first step that we hope will support future work in viewing OT from a fresh, technical perspective.

For completeness, below we briefly mention three other perspectives relevant to our research that we explored and but ultimately discarded.

2.4.1 Computational Organization Theory

The field of Computational Organization Theory (COT) "focuses on theorizing about, describing, understanding, and predicting the behavior of organizations and the process of organizing" (Carley and Wallace, 2001). COT researchers use models to describe a range of organizational characteristics as well as to understand real-world observations. Dozens of computational models have been developed including DYCORP and OrgAhead (Ashworth and Carley, 2007).

Lin and Carley (1995) created the DYnamic Computational ORganizational Performance (DYCORP) framework that contrasts organizations with different designs and operating environments. Its purpose is to analyze four design elements: organizational structure, resource access structure, organizational procedures, and agent style. These elements are variable in order to simulate different organizational stresses. "Using this framework the researcher can generate a series of precise, and therefore refutable, predictions about the relationships among design, task, stress, and performance" (Lin and Carley, 1995). Unfortunately, DYCORP is designed to only examine four specific organizational structures and therefore is not suitable for our research objectives.

OrgAhead is another computational framework that models organizational learning and decision-making. It "is used to test various aspects of real life organizations, such as complexity in the task environment and constraints on structure and adaptability" (Lee and Carley, 2004). OrgAhead can only consider organizations with hierarchical structures and uses a Monte-Carlo style simulation which "evolves" the organizational structure using random perturbations to find an "optimal" solution (Lee and Carley, 2004). Measuring only hierarchical organizational structures limits the relevance of OrgAhead to our research objectives and therefore it was not used.

2.4.2 Network Science

Network Science spans multiple fields of study, including mathematics (through graph theory), sociology (through social networks), computer science (through algorithms for exploring networks), and engineering (through infrastructural networks); see Barabási (2016). Network science lends itself well for modeling organizational architectures and measuring characteristics (size, degree, density, connectedness, etc.) but not representing the complexities of organizational architectures as Albert's three dimensions describe. Alderson (2008) summarizes the advantages and disadvantages of a network science perspective for the study of complex systems, as viewed from the lens of operations research. Despite its multi-disciplinary nature which offers a variety of perspectives we omit network science discussions from our research.

2.4.3 The OODA Loop

Another perspective for understanding C2 is to consider it as a decision making process. First presented in 1981 by Colonel John Boyd at the Air War College and discussed in *Power to the Edge* by (Alberts and Hayes, 2003), the military C2 process incorporates the four fundamental functions of Observe, Orient, Decide, and Act. Popularly called the "OODA Loop," its four steps are a simple representation of the decision making cycle used at all levels of C2. Its simplicity resulted in its fast adoption across military forces.

Simple models commonly trade tractability for accuracy. The OODA Loop's four functions lack necessary detail in order for it to be an effective analytic tool. Efforts to make Boyd's work more descriptive include the work of Breton and Rousseau (2006) who propose a modular version ("M-OODA") that expands the "iterative and dynamic notions with feedback and feed-forward [loops]" (Breton and Rousseau, 2006). In 2008 they develop the Cognitive-OODA ("C-OODA") loop which enhances the previous model by "increasing its level of cognitive granularity" (Breton, 2008).

The variety of alternative models is evidence that no single decision making model can successfully capture the complex nature of C2.

2.5 *Team of Teams* — A Closer Look

In Iraq in 2003, the highly trained and well equipped JSOTF was unable to defeat the lesser equipped and trained AQI forces. In his book *Team of Teams*, retired Army General McChrystal describes the difficulties his task force faced as a result of having a traditional military architecture (McChrystal et al., 2015). The JSOTF was unable to anticipate and be pro-active in preventing AQI attacks as a result of the enemy's unpredictability and fast pace. Instead, the architecture restricted McChrystal's forces to a reactive posture.

Table 2.1 highlights characteristics of JSOTF and AQI from an OT perspective.

Table 2.1. Contrast of JSOTF and AQI. (Adapted from Daft, 2012)

JSOTF (Mechanistic)	**AQI (Organic)**
Centralized Structure	Decentralized Structure
Strict Hierarchy of Authority	Collaborative Teamwork
Specialized Tasks	Empowered Roles
Vertical Communication	Horizontal Communication
Many Rules, Formalized	Few Rules, Informal

After realizing his task force's fundamentally mechanistic architecture was ineffective against its unpredictable and agile enemy, McChrystal restructured the JSOTF's architecture and culture using concepts learned from *netwar*. Developed by John Arquilla and David Ronfeldt, *netwar*, "refers to an emerging mode of conflict (and crime) at societal levels, short of traditional military warfare, in which the protagonists use network forms of organization and related doctrines, strategies, and technologies attuned to the information age" (Arquilla and Ronfeldt, 2001). In its battle against AQI, the JSOTF had to first learn about its enemy's architecture. According to Arquilla and Ronfeldt (2001), non-state actors and terrorists like AQI, benefit from "numerous dispersed small groups using the latest communications technologies [that can] act conjointly across great distances."

Netwar organizations like AQI, are a network of highly connected and motivated fighters. "The organizational design is flat. . . there is no single, central leadership, command, or headquarters - no precise heart or head that can be targeted" (Arquilla and Ronfeldt, 2001). Such organizations have advantages over hierarchies because decision making authority is decentralized, thus allowing for "local initiative and autonomy" (Arquilla and Ronfeldt, 2001). Using the proposition, "It takes networks to fight networks" (Arquilla and Ronfeldt,

2001) and key traits of netwar, McChrystal reorganized his forces in the following ways.

1. *Teams*: McChrystal disbanded traditional military departmentalization and reorganized into cross-business teams. This structure supported creativity, communication, and accountability. Boundaries or procedures which disrupted or prevented communication and collaboration between teams were identified and removed.
2. *Information Sharing*: McChrystal consolidated multiple communication networks and demanded information be available throughout his task force. No person or team in the task force can retain specialized information. He insisted on full transparency and to this end conducted an Operations and Intelligence (O&I) briefing six times weekly to push command guidance and actionable intelligence to the edges of the task force with great speed and accuracy. By doing so, McChrystal created a “shared consciousness” which supports adaptability in fighting an unpredictable enemy (McChrystal et al., 2015).
3. *Liaison Officers*: Fundamental to McChrystal’s new task force structure are embedded liaison officers. Representatives from other organizations enhance collaboration and cooperation. Previous to McChrystal, the task force’s level of interaction and cooperation with other government agencies (i.e. CIA and NSA) did little to aid either organization (McChrystal et al., 2015). By carefully selecting quality liaison officers, McChrystal increased inter-agency trust, transparency, and cooperation.
4. *Empowered Execution*: To combat a fast paced enemy, leadership must make decisions in a timely manner. Instead of trying to control each decision in his task force, McChrystal empowered those “individuals and teams closest to the problem, armed with unprecedented levels of insights from across the network” with the “ability to decide and act decisively” (McChrystal et al., 2015). McChrystal ensured this decentralized decision-making culture supported the task force’s goals by clearly communication Commander’s Intent during O&I briefings (McChrystal et al., 2015).

By making the necessary changes to create a shared consciousness, extreme transparency, effective communication forums, and empowered execution, McChrystal’s task force became a “team of teams.” The new flatter, networked, and team-oriented architecture reflected “an organization within which the relationships between constituent teams resembled those between individuals on a single team” (McChrystal et al., 2015) and embodied netwar characteristics to better understand and combat AQI.

2.6 Why this Research?

In support of quantitatively telling the *Team of Teams* story from an operations research perspective, this research establishes a foundation for such quantitative discussions for measuring an organization's architecture's goodness-of-fit with its environment. To do so, we develop a framework for measuring how well an organizational architecture performs in a known environment while accomplishing a known objective. This framework, in the form of a simple game, imitates the basic processes of an organization and gives us a means to simulate various C2 architectures and contrast their performance. Using mathematical analysis, simulation, and Albert's three dimensions of organizational architecture we will explain why the JSOTF architecture originally failed against AQI and why McChrystal's organic "team of teams" concept was a better fit for its environment.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

3.1 Let's Play a Game

We seek a framework to quantitatively measure how well an organization's architecture achieves its mission in a given environment. As a first-principles approach to creating such a measurement construct, we seek a design which mimics organizational processes and yet is simple enough to explain and support mathematical analysis. The measurement framework we present is in the form of a game.

3.1.1 What is a Game?

According to Fullerton (2014), games contain five major elements:

Components. These are players and the objects in the game with which players interact. *Components* of board games are commonly dice or cards, whereas in football they are team members and the ball.

Space. The "world" in which the game is played. The *space* of board games like Monopoly are specifically designed boards, whereas the *space* of football is the playing field.

Goals. The objective the player(s) strive to achieve. *Goals* define what it means to win. In Monopoly, the objective is financial dominance, whereas in football both teams strive to accrue maximum points.

Mechanics. These are the actions players can do, or have done to them. During game-play, *mechanics* constrain player actions or movement, often to add difficulty to the game. In Monopoly, rolling dice determines which property a player moves to. In football, each team has four "downs" to advance a minimum of 10 yards.

Rules. The constraints which guide a player's decision making. *Rules* describe how the game is played and what actions are allowed to be taken in order to win. In Monopoly, a player can buy an available property or perhaps force rent to be paid if it is already owned. In football, penalties for rule violations ensure fair play (e.g., holding, pass interference) and player safety (e.g., roughing the kicker, targeting, and clipping).

3.1.2 Game Design Requirements

We seek a game: 1) That mimics the organizational processes of gathering, filtering, assembling, and delivering, 2) That can be played by a single player, 3) In which an organization with additional players should perform better, 4) In which interactions and communication between players can be manipulated to represent varying types of organizational architectures, and 5) That can be played in different environments and with varying objectives.

Using Fullerton's five game elements, we describe the characteristics we seek for a game that can be used to quantify an organization's effectiveness under various circumstances.

Components. The *components* of our game are the players, who represent the members in an organization, and resources the players use.

Space. Our game's *space* is not associated with a geographic location, specialized game board, or type of playing field. In terms of our game, *space* represents an organization's operating environment. "*Organizational environment* is defined as everything that exists outside the boundary of the organization and has the potential to affect all or part of the organization" (Daft, 2001). The environment also incorporates an important temporal element (according to Morgan, 2006) which is its "degree of stability or change". We require the ability to change the environment simulated in our game in order to test architecture performance under varying circumstances.

Goals. The *goal* of our game is for an organization to complete a specific objective. We wish to measure an architecture's performance in completing its objective by recording the time it takes to do so.

Mechanics. The game's *mechanics* define who can communicate/collaborate with who and formal reporting relationships between members. These define the organizational architecture being examined.

Rules. We require a game with *rules* that represent simple 'if-then' player decisions. Avoiding complex decision making situations keeps the game understandable and supports mathematical tractability. Game rules are adjustable to control player actions in support of modeling various organizational architectures.

3.2 Choosing the "Correct" Game

Using these design attributes we originally sought to find an existing, recognizable game to serve as the basis of our framework. Choosing a pre-existing and familiar game helps prevent confusion and avoid more lengthy discussions. In our search, we considered the following potential games for exploration.

Candidate 1: Memory

The childhood memorization game "Memory" inspired our first design iteration. In this game, n number of cards (consisting of $\frac{n}{2}$ number of pairs) are placed face down randomly in a grid. During each turn, a player selects two cards in attempt to match the second card with the first. If the two cards do not match they are returned face down to their original positions. When a match is made, the two cards are removed thus reducing the remaining number of cards. The goal of the game is to assemble as many matches as possible. This game is attractive because it is feasible to scale up the game and number of players but the complexities of simulating player memory and the possibility for involving partial matches led us to rule out this game.

Candidate 2: Bingo

The second game we examined is "Bingo." In this game, the player or players receive numbers randomly from a distribution and must match them to a game board. This game supports complete matches but complications arise with the variations of how a game can end (e.g., matching a row vs. column). Additionally, we saw no added benefit beyond a linear increase in performance when playing with more than one player. For these reasons we continued our search elsewhere.

Candidate 3: Number Matching

A hypothetical number matching game was our next attempt. In this game, the player or players receive numbers randomly and they attempt to match these numbers with a "target sequence" (of numbers). This game received no additional consideration because we perceived no increase in performance from having any architecture more elaborate than a simple two layer hierarchy. Attempts to add extra rules and constraints to make the game more challenging was avoided because adding unnecessary complexities can be considered by readers as intentionally contrived in order to achieve a desired research outcome.

Candidate 4: "Fishing"

A modification to the mechanics of Candidate 3 was examined in which players had different probabilities of receiving random numbers from a distribution. The intent of this modification was to model organizational workers with different levels of skill and experience. A senior, more skilled worker could "fish" for numbers within the environment with greater success that a junior, less skilled worker. This modification added unnecessary mathematical complications and was not supportive of our research objectives.

3.3 Our Game: Word Matching

The chosen framework for our research simulation is a word matching game. Using Fullerton's lexicon we present the game's characteristics:

Components. The components of the game are simply the players and resources available to each player. Each player represents one member in an organization.

Space. The resources come from the game's space (an organization's environment) which we implement using a distribution of letters A through Z. During every turn, each player receives one letter from the distribution which we can vary, thus changing the organizational environment in which we examine the performance of a specific architecture.

Goals. The overall goal of the game is for an organization to collect and assemble the required letters to match a specific objective. The objective can be partitioned into small tasks and assigned to specific members in the organization. Each player is assigned a primary (and possibly a secondary) task which they strive to accomplish using the letters they receive. Like the environment, we can vary the organization's objective in order to measure an architecture's performance under different stresses.

Mechanics. The game's mechanics are defined by the specific organizational architecture being examined. Each architecture will have different connections between members, reporting relationships, and collaborative processes. We can change the game's mechanics in order to analyze different architectures.

Rules. Game rules direct player actions. They dictate what each player will do each turn according to the architecture being examined. The number of discrete simulation time steps it takes the organization to match its objective is the output (score) of the game.

3.4 Example Game Play

Next, we demonstrate how the word matching game is played and how the game measures the performance of an organizational architecture. For this example we examine the simplest possible architecture; a single person playing the game. This player's objective is the word "TEACH." Figure 3.1 illustrates this player receiving letters from the distribution one at a time. When a T, E, A, C, or H is received, the player retains the letter, otherwise the letter is discarded. The player continues receiving letters from the environment and repeats the filtering action. Once the player has enough of each letter to match the objective word, the player does so, thus completing organization's objective, and ending the game.

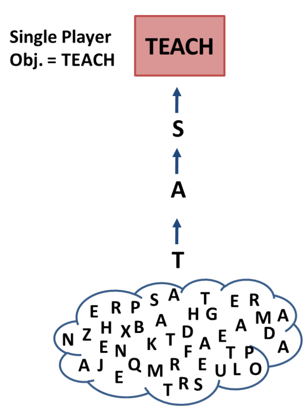


Figure 3.1. Single Person Architecture

3.5 Architectures to be Studied

Next, we identify the organizational architectures we seek to explore via mathematical analysis and simulation in Chapter 4. Our goal is to understand how resilient an architecture is to changes in the organization's objective and/or environment. Before presenting the architectures we must clarify our terminology:

Worker. A worker is a player in the game that receives an input of letters from the environment (distribution of letters). We illustrate workers using a red rectangle. Each is assigned a task (sub-component of the organization's overall objective), annotated in the rectangle. Workers can only communicate or collaborate with other players if connected by a linkage in the architecture. Workers have no subordinates.

Manager. A manager is a player in the game connected to one or more subordinate workers. Managers are illustrated with blue rectangles and do not receive an input of letters from the environment. The manager's task is to collect completed tasks from workers and match the organization's overall objective, depicted in the rectangle.

Using this lexicon and the word matching game, we present the architectures we will examine in our analysis:

3.5.1 Single Worker

As illustrated in Figure 3.1, a one person organization has the simplest architecture possible. This is our base case from which we begin our analysis.

3.5.2 Two Workers Playing Independently

The next architecture we consider consists of two workers who cannot communicate or collaborate in any way while playing the game. Each worker has the same task but the organization's objective is satisfied when either player matches it completely. Each worker receives letters independently and chosen randomly from the distribution. This architecture is illustrated in Figure 3.2.

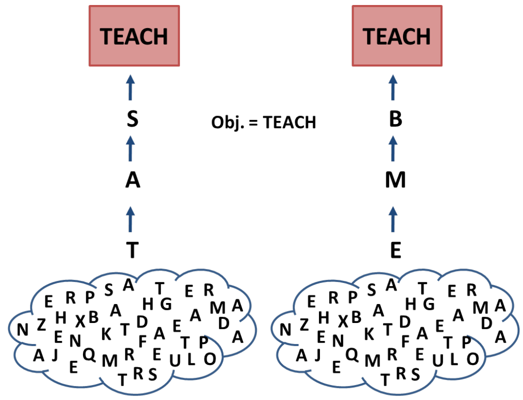


Figure 3.2. Two Workers Playing Independently

3.5.3 Two Workers Cooperating

Next, we examine an architecture which allows two workers to play while sharing letters. Unlike two independent players, this organization's objective is complete when the collective letters of both players are sufficient to match the assigned objective. As shown in Figure 3.3, this architecture connects two workers allowing them to play the game cooperatively.

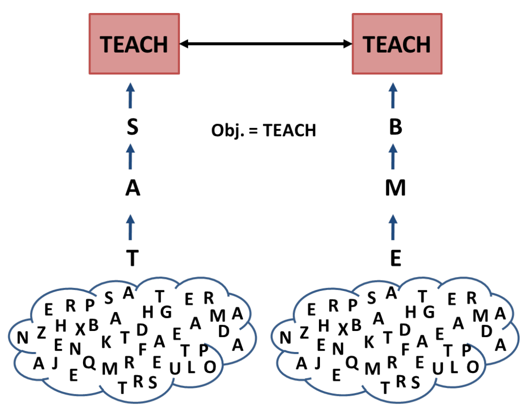


Figure 3.3. Two Workers Playing Together

3.5.4 Simple Hierarchy

We next consider a simple hierarchy consisting of two workers who are connected by a manager. This architecture partitions the organization's objective into smaller tasks. As illustrated in Figure 3.4, one manager collects completed tasks from the two workers in order to match the organization's overall objective. Additionally, the manager can distribute unused letters to other players if needed.

NOTE: *Beginning with Figure 3.4 all illustrations will show only the organizational architecture and omit any graphics representing the distribution of letters (environment).*

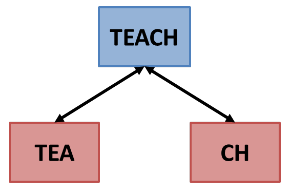


Figure 3.4. Example Simple Hierarchy

3.5.5 Larger Hierarchy

We consider next, larger hierarchies and explore why this architecture is well-suited for organization's with singular objectives and that operate in a stable environment. Organizations with mechanistic-style hierarchies (as depicted in Figure 3.5) focus on efficiency as their measure of performance. We show that efficiency is obtained at a cost of robustness and flexibility to change.

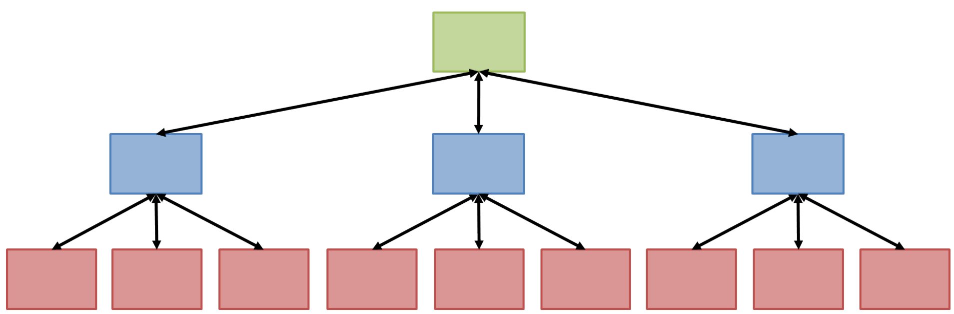


Figure 3.5. Example Larger Hierarchy

3.5.6 Teams

Team architectures are the fundamental element of Contingency Theory. Teams consist of multiple members in an organization who collaborate to complete a given task or objective. As illustrated in Figure 3.6, team architectures foster creative thinking and innovation because the denser number of connections between team members support communication and information sharing. Organizations utilizing team architectures are characterized by flexibility and robustness to changes in the organization's objective or environment.

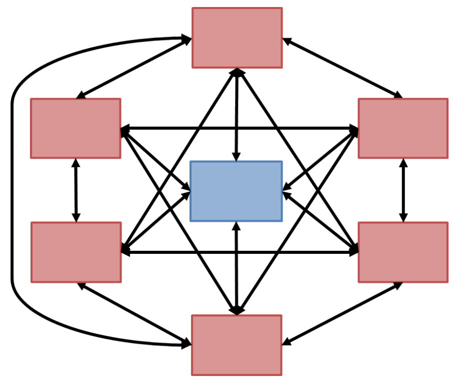


Figure 3.6. Example Team Architecture

3.5.7 Team of Teams

As teams grow larger the number of connections between members can increase rapidly. Since communication and collaboration has an associated burden on players, larger densely connected organizations can reach a peak level of performance. To help tell the story of McChrystal's *Team of Teams*, we consider architectures which connect multiple teams together as illustrated in Figure 3.7. Since it is not feasible in large organizations for every member to connect with every other member, teams assigned to specific tasks are connected to other teams.

In Chapter 4 we mathematically analyze the more simple architectures discussed in this section. We use this analysis to predict the performance of these architectures when tested using our word matching game simulation. We start with our base case architecture and then incrementally add complexity.

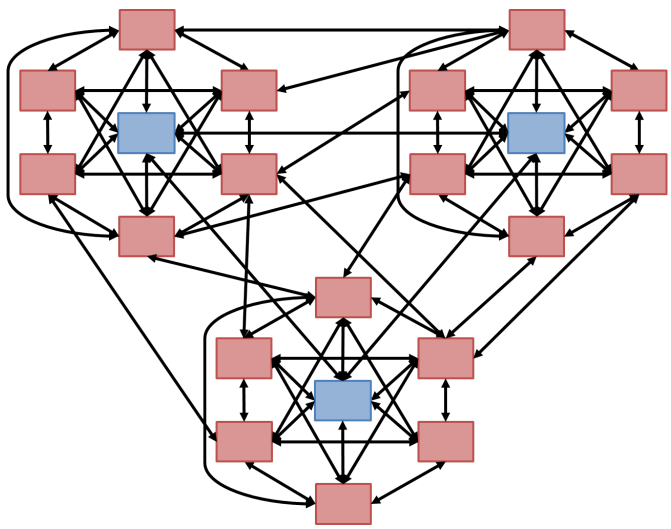


Figure 3.7. Example Team of Teams Architecture

CHAPTER 4:
Analysis

4.1 Introduction

In this chapter, we use a Discrete-Time Markov Chain as a mathematical framework to support analysis of our word matching game. Specifically, we represent play in any specific game as a series of random transitions from one *state* to another, starting with an initial state and concluding when the system reaches a terminal (or trapping) state. At every point in the game, we assume the probability of the next transition depends only on the current state (i.e., the process is memoryless). Time advances in discrete steps and the *state space* for the game is the set of all possible states.

4.1.1 Notation and Mathematical Setup

Let $\mathcal{S}$ represent the state space of the system, assuming without loss of generality that there are $|\mathcal{S}| = n$ total states. We partition the set $\mathcal{S}$ into two subsets, where, $\mathcal{T} \subset \mathcal{S}$ is the set of transitive states, and $\mathcal{U} \subset \mathcal{S}$ is the set of trapping states, with $\mathcal{T} \cap \mathcal{U} = \varnothing$ and $\mathcal{T} \cup \mathcal{U} = \mathcal{S}$. Let i (alias j) index $\mathcal{S}$. Additionally, we assume $|\mathcal{T}| = m < n$ and that states in $\mathcal{S}$ are ordered such that,

$$i \in \{\underbrace{1, 2, \ldots, m}_{\text{transitive}}, \underbrace{m+1, \ldots, n}_{\text{trapping}}\}.$$

Let P be the associated transition matrix of the system, where elements P_{ij} represent the probability of transitioning from state i to state j. Given the assumed ordering above, we rewrite the transition matrix P as follows:

$$P = \begin{bmatrix} T & T_0 \\ 0 & I \end{bmatrix}. \tag{4.1}$$

Here, the submatrix T represents the transitions between transitive states, and the submatrix T_0 represents transitions from transitive to trapping states. By construction, there are no transitions from trapping states to transitive states, and once in a trapping state, the system stays there permanently.

Let τ_i represent the expected number of transitions from state i until arriving in a trapping state. We wish to calculate values for $\tau_1, \tau_2, \ldots, \tau_m$ (note, $\tau_{m+1}, \ldots, \tau_n$ are each equal to zero by construction). This is done by solving the following system of equations:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_m \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \begin{bmatrix} & & \\ & T & \\ & & \end{bmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_m \end{pmatrix}, \tag{4.2}$$

where T is a matrix of size $m \times m$.

In order to use this framework for our word matching game, we define the state space in terms of the number of letters that have been obtained. The transitions between states depend in general on the number of players and connections between them as determined by the architecture being examined. Specific examples follow. However, in general we measure the performance of a given architecture for a game in terms of the expected time (number of discrete simulation steps) to complete a task, which translates directly to the value of τ for some initial state.

4.1.2 Simulation

We complement our mathematical analysis of the word matching game using a Monte-Carlo style simulation coded in the Python programming language (Rossum and Drake, 2003). A single run of the simulation returns the time for the organization to complete its objective. For each architecture of interest, we run the simulation 10,000 times and calculate the sample mean plus or minus a 95% confidence interval half-width. Below we discuss the simulation using the lexicon of Fullerton (2014).

Components

We define a Python class-object called `Agent` to represent *workers* and *managers* in an organization. Every agent is assigned a *primary task* and has an optional *secondary task*. For a worker, a primary task could be a letter, word, or letter sequence that they must match. A manager's primary task is to assemble components provided by workers to match the organization's objective. Secondary tasks are assigned according to the architecture being examined (e.g., to share letters needed by other workers). Each agent has a queue from which they get their next letter or item. Depending on the architecture being measured, a worker's queue is populated with letters that are either sampled from the distribution or shared by other workers. The manager's queue can contain components of the objective that were matched by workers or unused letters to share with other workers. In each discrete step of the simulation, the number of items an agent can process is limited by the *work capacity* for that respective type of agent. We consider the specific case where a worker can process only one item from their queue, whereas a manager can process two items. Additionally, every agent has a `WorkSpace` in which letters needed for the primary are retained, until they are removed after a successful match is made.

Space

We define the space or environment of the game by a distribution of letters. During each turn every worker receives one random letter from the distribution. The first letter distribution we present is directly proportionate with the frequency of letters found in 40,000 words from the English dictionary (Cornell Department of Mathematics, 2004). The frequency of each letter divided by the sum of all letters yields each letter's probability. Appendix B lists the frequency and probability for each letter in the Cornell letter distribution.

Goals

The organization's objective can be a single letter, sequence of letters, or word. Each run of the simulation terminates when the organization obtains the minimum quantity of letters needed to match the objective.

Mechanics

The simulation tests how well a specific organizational architecture performs in completing a given objective within a specified environment. An architecture is defined in terms of the connections between agents. We represent the connections for each `Agent` in terms of three lists (Parents, Children, and Peers) which define its position and connectivity within an organization. An agent's `ParentsList` defines "vertical" connections with agents who are superior, if applicable. Similarly, the `ChildrenList` connects agents with subordinate agents who are lower in the organization's hierarchy. The `PeersList` connects agents "horizontally" to other agents according to the architecture being examined.

Rules

The rules of the simulation determine agent actions. When a worker receives a letter that is needed for the primary then the worker adds it to the `WorkSpace`. Depending on the architecture being examined, non-matching letters are discarded or shared. Information sharing within an organization can be represented by sharing duplicate or unneeded letters with other workers (if they are connected) or via a manager.

We implement the Monte-Carlo style simulation using the following logic:

```
initialize time = 0
while not objective_matched:
    for each worker_agent:
        sample letter from distribution; add to queue
        pop letter from queue
        process letter
    for each manager_agent:
        for each turn in work_capacity:
            pop item from queue
            process item
    increment time
return time
```

We repeat this simulation 10,000 times and calculate the sample mean and sample 95% confidence interval half-width.

4.2 Single Player

The first architecture we analyze is our base case, comprised of only one worker who must match an objective (letter, word, or an assigned sequence of letters). In this section we present mathematical analysis of the base case to set the stage for more complicated work in later sections.

4.2.1 Matching a Single Letter

First, we consider an elementary example in which a worker's primary task is to match only a single letter, the letter "A". We represent our state space using a single element $\mathcal{S}_A \in \{0, 1\}$ where 0 represents the state where the letter "A" has not yet been matched, and 1 represents the state when the letter "A" has been matched. Figure 4.1 illustrates the Markov-Chain for the base case. The worker begins in state $\mathcal{S}_A = 0$. The worker remains in state 0 until an "A" is received. Using the Cornell letter distribution, the probability the next letter the worker receives is an "A" ($P_{0,1}$) is $P_A = 0.08124$. The probability the worker remains in state $\mathcal{S}_A = 0$ is $P_{0,0} = 1 - P_{0,1} = 0.91876$.

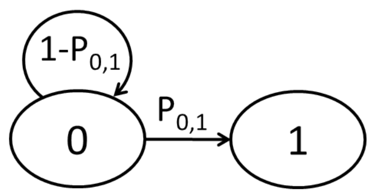


Figure 4.1. Markov Chain of Single Player Matching Single Letter

Somewhat trivially, the primary task is achieved when the system reaches state $\mathcal{S}_A = 1$. Let $E[N_0]$ be the expected number of letters (equivalently, time steps) the worker should see on average before completing the primary task, if starting at state $\mathcal{S}_A = 0$. In this simple case, we calculate this expected value using,

$$\begin{aligned}
E[N_0] &= 1 + P_{0,1}E[N_1] + (1 - P_{0,1})E[N_0] \\
& \quad \text{where } E[N_1] = 0 \\
E[N_0] &= 1 + (1 - P_{0,1})E[N_0] \\
E[N_0] - (1 - P_{0,1})E[N_0] &= 1 \\
P_{0,1}E[N_0] &= 1 \\
E[N_0] &= \frac{1}{P_{0,1}}. \qquad (4.3)
\end{aligned}$$

Using Equation 4.3, we calculate,

$$E[N_0] = \frac{1}{P_{0,1}} = \frac{1}{0.08124} = 12.3095.$$

This simple result is intuitive because system transitions are governed by a Bernoulli random variable. Each letter the agent receives can only move the system into two possible states: 1) the letter did not match the objective, or 2) the letter matched the objective. Like flipping a coin, the agent's objective is complete as soon as the first match is made, otherwise the system continues.

Using our Python implementation of the word matching game, we simulate a single worker (base case) architecture with three different single letter primary tasks. Table 4.1 compares calculated answers (using Equation 4.3) with simulation results.

Table 4.1. Single Worker, Single Letter Calculated Results vs. Simulated

Task:	P_x:	Calculated Value:	Simulated Value:
A	$P_A = 0.08124$	12.3095	12.3797 ± 0.2291
B	$P_B = 0.01489$	67.1466	67.2827 ± 1.3052
C	$P_C = 0.02711$	36.8810	36.9857 ± 0.7132

By beginning our analysis with the simplest architecture and confirming the mathematical analysis and simulation results match, we verify our setup, thus establishing a foundation which we will incrementally build upon to examine more advanced architectures.

4.2.2 Matching Multiple Letters

Now lets assume a single worker's task is to match the two letter sequence "AB." Figure 4.2 shows the Markov-Chain for this case. The state space is now defined by a pair of values (S_A, S_B) where $S_A \in \{0, 1\}$ and $S_B \in \{0, 1\}$. Thus, the number of states has increased from two to four.

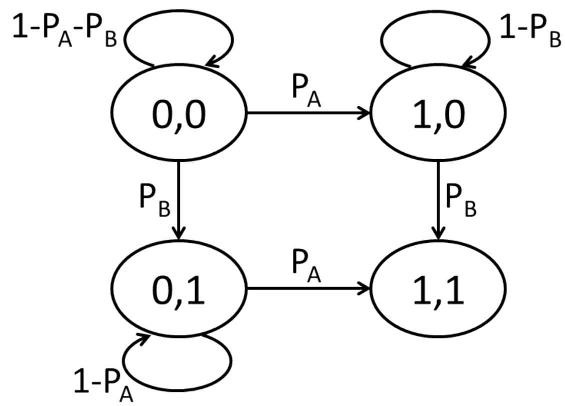


Figure 4.2. Markov Chain of Single Worker with Task "AB"

Using Figure 4.2, we derive the analytic formula for $E[N_{00}]$ as follows,

$$\begin{aligned}
E[N_{00}] &= 1 + P_A E[N_{10}] + P_B E[N_{01}] + (1 - P_A - P_B)E[N_{00}] \\
&\quad \text{where } E[N_{01}] = 1 + (1 - P_A)E[N_{01}] \Longrightarrow E[N_{01}] = \frac{1}{P_A} \\
&\quad \text{where } E[N_{10}] = 1 + (1 - P_B)E[N_{10}] \Longrightarrow E[N_{10}] = \frac{1}{P_B} \\
E[N_{00}] &= 1 + P_A\Big(\frac{1}{P_B}\Big) + P_B\Big(\frac{1}{P_A}\Big) + (1 - P_A - P_B)E[N_{00}] \\
&= \frac{1}{P_A + P_B} + \frac{P_A}{P_A + P_B}\Big(\frac{1}{P_B}\Big) + \frac{P_B}{P_A + P_B}\Big(\frac{1}{P_A}\Big). \qquad (4.4)
\end{aligned}$$

Using the Cornell letter distribution (in which $P_A = 0.08124$ and $P_B = 0.01489$) and Equation 4.4 we calculate,

$$\begin{aligned}
E[N_{00}] &= \frac{1}{P_A + P_B} + \frac{P_A}{P_A + P_B}\Big(\frac{1}{P_B}\Big) + \frac{P_B}{P_A + P_B}\Big(\frac{1}{P_A}\Big) \\
&= \frac{1}{0.0961} + \frac{0.08124}{0.0961}\Big(\frac{1}{0.01489}\Big) + \frac{0.01489}{0.0961}\Big(\frac{1}{0.08124}\Big) \\
&= 69.0536.
\end{aligned}$$

We simulated a single worker with three different two letter tasks. Table 4.2 supports the analytic calculations shown above.

Table 4.2. Single Worker, Multiple Letter Calculated Results vs. Simulated

Task:	Calculated Value:	Simulated Value:
AB	69.0536	68.7626 ± 1.2761
SC	41.6819	41.8585 ± 0.6780
JL	970.3319	969.9242 ± 18.7888

The mathematical calculations and simulation results in Section 4.2 are consistent, thus supporting the accuracy of the word matching game simulation as a measurement construct for more complicated organizational architectures as we continue research.

4.3 Multiple Independent Players

In this section, we examine the performance of organizational architectures with two or more workers who operate independently to achieve identical primary tasks (as illustrated in Figure 3.2). Workers play the game in parallel and without any interaction, communication, or coordination. The organization's objective is met as soon as one of the multiple workers completes the assigned primary task.

4.3.1 Matching a Single Letter

Figure 4.3 shows a Markov-Chain for two workers, each with the same primary task (match "A"). We denote the state space now by (S^1, S^2), where $S^1 \in \{0, 1\}$ and $S^2 \in \{0, 1\}$ indicate whether Worker 1 or Worker 2, respectively, has matched the letter. State (0,0) is the starting state, and states (1,0), (0,1) and (1,1) are all absorbing states because the organization's objective is complete as soon as either worker gets their first "A" or in the case where both workers get an "A" on the same turn.

Using figure 4.3, we derive $E[N_{00}]$ as follows,

$$
\begin{aligned}
E[N_{00}] &= 1 + (1 - P_A)^2 E[N_{00}] + [1 - (1 - P_A)^2](0) \\
E[N_{00}] &= 1 + (1 - 2P_A + P_A^2) E[N_{00}] \\
[1 - (1 - 2P_A + P_A^2)] E[N_{00}] &= 1 \\
E[N_{00}] &= \frac{1}{2P_A - P_A^2}. \qquad (4.5)
\end{aligned}
$$

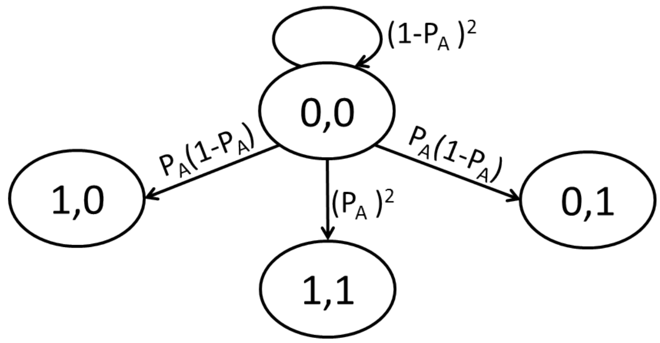


Figure 4.3. Markov Chain of Two Workers with Same Single Letter Task

Equation 4.5 can be generalized for k number of players as follows,

$$\begin{aligned} E[N_{00}] &= 1 + (1 - P_A)^k E[N_{00}] + 0 \\ [1 - (1 - P_A)^k] E[N_{00}] &= 1 \\ E[N_{00}] &= \frac{1}{1 - (1 - P_A)^k}. \end{aligned} \tag{4.6}$$

Using the word matching game simulation we measure the performance of the architectures with two or more workers who work independently to accomplish their task. Table 4.3 shows the calculated expected values using Equation 4.6 compared to simulation outputs. The results support the common sense notion that two or more workers should complete a single letter primary task faster than one.

Table 4.3. Results of Multiple Independent Workers with Task = "A"

# of Workers:	1	2	3	4
Calculated:	12.3092	6.4152	4.4552	3.4787
Simulated:	12.3797 ± 0.2291	6.3986 ± 0.1150	4.5630 ± 0.0810	3.4597 ± 0.0573

4.3.2 Two Players, Matching Multiple Letters

Next, we increase the difficulty of the primary task from a single letter to two letters. This increases the state space from four to 16 because we must use four digits to keep track of system states. To do so, we let $\mathcal{S}_i^k$ = state of player k on letter i. If the organization has two players and two letters ("AB") then $\mathcal{S}_i^k = (\mathcal{S}_A^1, \mathcal{S}_B^1, \mathcal{S}_A^2, \mathcal{S}_B^2)$. For example, $\mathcal{S}_{0,1,0,0}$ represents the system state in which worker #1 has matched a “B” but not an “A,” and worker #2 has matched neither. Using this notation, Table 4.4 is the transition submatrix T of matrix P.

We solve the system of equations defined by T using the statistical computing language R (R Core Team, 2016), and obtain $E[N_{0000}] = \mathcal{T}_1 = 36.4587$. This analytic result is consistent with the simulation output for this architecture, equal to 36.6198 ± 0.6240.

As the number of workers or size of the objective increase, the size and complexity of an organization’s transition matrix can very quickly become too complicated to manually produce. As shown in Table 4.4, two workers with a two letter primary task requires a system with 16 states. For each additional worker or letter added to the organization or primary task, respectively, the system’s state space doubles. For example, the dimensions of the P matrix for an architecture with three independent workers each with task “ABC” is 64 x 64. Setting up the transition matrix becomes tedious as organizations grow beyond trivial sizes. For this reason, when examining more complicated architectures in subsequent sections we rely more heavily on simulation analysis.

The simulation outputs in Table 4.6 show the impact of adding more independent workers to an architecture with objective “AB.” The results are intuitive. With more workers, the expected time to complete the task decreases. However, if all that matters is completing this task once, then any partial progress of the other workers is wasted. One way to mitigate this wasted work is for the workers to coordinate their activities to work together instead of independently. Next, we explore architectures with connections between workers that allow communication or cooperation.

Table 4.6. Simulation Results of Independent Workers with Task = “AB”

# of Workers:	2	3	4
Expected Value:	36.6198 ± 0.6240	25.8209 ± 0.4025	20.9587 ± 0.3104

Table 4.4. Transition Submatrix T for Two Independent Workers (Each with Primary Task of "AB")

	(0, 0, 0, 0)	(0, 0, 0, 1)	(0, 0, 1, 0)	(0, 1, 0, 0)	(0, 1, 0, 1)	(0, 1, 1, 0)	(1, 0, 0, 0)	(1, 0, 0, 1)	(1, 0, 1, 0)
(0, 0, 0, 0)	$(1-P_A-P_B)^2$	$P_B(1-P_A-P_B)$	$(1-P_A-P_B)P_A$	$P_B(1-P_A-P_B)$	$(P_B)^2$	P_BP_A	$P_A(1-P_A-P_B)$	P_AP_B	P_A^2
(0, 0, 0, 1)	0	$(1-P_A-P_B)(1-P_A)$	0	0	$P_B(1-P_A)$	0	0	$P_A(1-P_A)$	0
(0, 0, 1, 0)	0	0	$(1-P_A-P_B)(1-P_B)$	0	0	$P_B(1-P_B)$	0	0	$P_A(1-P_B)$
(0, 1, 0, 0)	0	0	0	$(1-P_A)(1-P_A-P_B)$	$(1-P_A)P_B$	$(1-P_A)P_A$	0	0	0
(1, 0, 0, 0)	0	0	0	0	0	0	$(1-P_B)(1-P_A-P_B)$	$(1-P_B)P_B$	$(1-P_B)P_A$
(0, 1, 0, 1)	0	0	0	0	$(1-P_A)^2$	0	0	0	0
(1, 0, 1, 0)	0	0	0	0	0	0	0	0	$(1-P_B)^2$
(0, 1, 1, 0)	0	0	0	0	0	$(1-P_A)(1-P_B)$	0	0	0
(1, 0, 0, 1)	0	0	0	0	0	0	0	$(1-P_B)(1-P_A)$	0

Table 4.5. Transition Submatrix T for Two Cooperating Workers (Each with Primary Task of "AB")

	(0,0,0,0)	(0,0,0,1)	(0,0,1,0)	(0,1,0,0)	(0,1,0,1)	(1,0,0,0)	(1,0,1,0)
(0, 0, 0, 0)	$(1-P_A-P_B)^2$	$(1-P_A-P_B)P_B$	$(1-P_A-P_B)P_A$	$P_B(1-P_A-P_B)$	$(P_B)^2$	$P_A(1-P_A-P_B)$	$(P_A)^2$
(0, 0, 0, 1)	0	$(1-P_A-P_B)(1-P_A)$	0	0	$P_B(1-P_A)$	0	0
(0, 0, 1, 0)	0	0	$(1-P_A-P_B)(1-P_B)$	0	0	0	$P_A(1-P_B)$
(0, 1, 0, 0)	0	0	0	$(1-P_A)(1-P_A-P_B)$	$(1-P_A)P_B$	0	0
(1, 0, 0, 0)	0	0	0	0	0	$(1-P_B)(1-P_A-P_B)$	$(1-P_B)P_A$
(0, 1, 0, 1)	0	0	0	0	$(1-P_A)$	0	0
(1, 0, 1, 0)	0	0	0	0	0	0	$(1-P_B)^2$

4.4 Multiple Cooperating Players

We now examine the performance of architectures with two or more workers who are connected and can cooperate to complete the organization's objective. In these architectures, workers receive letters by sampling independently from the distribution, but can use letters shared by other workers. In terms of our word matching game simulation, we implement the sharing of letters in this section by simply combining each agent's `WorkSpace` list into one. In other words, when one worker matches an "A," all workers have an "A" in their `WorkSpace`. The architecture matches its objective when the combined `WorkSpace` contains the required letters. We use this over-simplified letter sharing rule in this section only until we introduce the concept of costs associated with cooperation.

4.4.1 Matching a Single Letter

Recall the two-person architecture (illustrated in Figure 3.3) where two workers are connected and can share letters they receive. Let the architecture's objective be to match "A." In the case where there are no *cooperation costs* for players to share letters (more on this below), then this architecture will perform equivalently to two independent (non-connected) workers. This is because, for either architecture, its system ends as soon as any worker gets an "A." This means that for a single letter objective, the expected performance for two or more independent workers (Equation 4.6) is the same as two or more cooperating workers. Using the word matching simulation we confirmed that both systems performed equally. Only in the case of single letter objectives does the expected value of k independent workers equal k cooperating workers. As soon as an organization's objective is larger than a single letter, the advantage of cooperation between multiple workers is observable.

4.4.2 Matching Multiple Letters

As an organization's objective grows in size, cooperation among workers improves organizational performance. Consider the instance of two workers who cooperate to match a two-letter primary task ("AB"). We again define the state space as $(\mathcal{S}_A^1, \mathcal{S}_B^1, \mathcal{S}_A^2, \mathcal{S}_B^2)$, and Table 4.5 displays the submatrix T for two cooperating workers with primary task of "AB." Compared to Table 4.4, the new transition submatrix is smaller because the system representing two cooperating workers has two additional trapping states. Specifically, because a trapping state is reached as soon as a minimum of one "A" and one "B" are matched by the two workers, the states (0,1,1,0) and (1,0,0,1) are now trapping states.

Using the updated submatrix T, we solve the system of equations numerically using R and calculate $E[N_{0000}] = \mathcal{T}_1 = 34.7854$. This is consistent with our simulation results for two cooperating players of 34.8019 ± 0.6404, thus supporting our statement that cooperation among workers improves performance.

To examine the performance of more than two cooperating workers, we rely on simulation. Table 4.7 shows a marked improvement in an organization's performance when its architecture connects workers, thus allowing cooperation. We show that adding more cooperating workers improves an organization's performance, but does the same statement remain true with larger numbers of workers? In the real-world, organizations have restraints on the size of their workforce and rarely can every worker be connected and cooperate. This is because cooperation comes at a cost to the organization.

Table 4.7. Simulation Results of Cooperating Workers with Task = "AB"

# of Workers:	2	3	4
Expected Value:	34.8019 ± 0.6404	23.2637 ± 0.4259	17.7431 ± 0.3175

Introducing Costs of Cooperation.

Cooperation can only occur between workers who are connected. In the real-world, these connections represent communications paths, interaction processes, or any formal means in which workers communicate. Unfortunately, time spent communicating is less time spent working toward accomplishing an assigned task. Stated differently, cooperation comes at a cost or expense to the organization's performance.

An organization with workers who are not connected does not spend time communicating (because they can't) and those workers have zero costs associated with cooperation. Unfortunately, a disconnected architecture fails to benefit from the performance advantages gained when workers are connected and cooperate. At the other extreme, an organization in which every worker is connected to every other worker has total sharing across the workforce. In this architecture, performance can suffer if workers are so overwhelmed sharing and receiving information that they can't perform their assigned duties.

We present Figure 4.4 to demonstrate the cost of cooperation even in a very small organization. The depicted architecture shows two workers who are connected and can share letters in order to match the organization's objective of "AB." Worker 1 has the primary task of matching "A" and the secondary task of sharing "B." Worker 2's tasks are opposite.

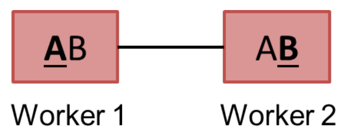


Figure 4.4. Two Connected Workers with Different Primary and Secondary Tasks

Each worker only gets one letter per simulation time-step. The letter is either a new letter from the distribution or it is a letter that was shared by the other worker. This means that by receiving and matching a shared letter, that worker is not exposed to a new letter from the distribution on that turn. For example, if Worker 2 gets an "A" then it is shared with Worker 1, who will use it instead of a fresh letter on the following simulation time-step.

Given the architecture shown in Figure 4.4 and the frequency of "A" compared to "B" in the Cornell letter distribution, we expect Worker 2 to more frequently share an "A" with Worker 1 than Worker 1 will share a "B" with Worker 2. This means that Worker 1 will receive fewer new letters from the distribution thereby reducing the number of chances to get the rarer letter ("B"), thus decreasing the overall performance of the organization. Using our word matching game simulation we confirmed this assertion with a result of $E[N_{0000}] = 36.2539 \pm 0.6664$ compared to the simulation result of $E[N_{0000}] = 34.8019 \pm 0.6404$ shown in Table 4.7.

Consider possible simulation results if repeated for larger organizations of 3, 4, or more fully connected workers. If a worker who is connected to multiple other workers receives a letter that is to be shared, to whom do they send it? When this happens in such architectures, now the worker must face a choice. The worker must choose which other worker to share the letter with and the organization's overall performance potentially depends on their decision. Inconsistencies in that decision can potentially result in inconsistent performance. Furthermore, figuring out how to make that decision takes time away from work the worker might otherwise be doing (receiving and attempting to match new letters from the distribution). A simple solution to this situation is to connect the worker to only a single person to whom they share the letter (i.e, the manager), like in a hierarchy. This keeps the worker's job simpler, and ensures consistency.

Before exploring architecture types like hierarchies that take advantage of selectively connecting workers, we emphasize that all results previous to this subsection do not consider costs associated with cooperation. The next section will explore trivial hierarchies and continue to ignore the concept of cooperation having an associated cost until we reintroduce it in Section 4.6.

4.5 Hierarchies

Hierarchical structures are common in organizations because they define positional relationships among members and break larger organizational objectives into separate or specialized tasks. They are often characterized by architectures in which large objectives are broken down into smaller tasks that can be accomplished efficiently.

We begin our discussion of hierarchies with a base case hierarchical architecture consisting of one manager and two workers, as shown in Figure 3.4. We examine this architecture first by assigning it the objective "AB" in order to compare its performance with previously discussed architectures. In this base case hierarchy, one worker is assigned the task of matching "A" and the other worker "B." The organization achieves its objective when the manager has received at least one "A" and one "B."

In this hierarchy, the manager's only action is to receive completed tasks from subordinates. Later, in Section 4.5.2 we examine a modification in which the manager can share letters among workers.

4.5.1 Base Case Hierarchy - One Way Vertical Communication

Let the pair of values $(\mathcal{S}_A, \mathcal{S}_B)$ define the state space, where $S_A \in \{0, 1\}$ and $S_B \in \{0, 1\}$. Each value in the pair is 0 or 1, representing the presence or absence of a letter match at the manager's level at that state space, respectively. Figure 4.5 illustrates the Markov-Chain for the base case hierarchy. From the view of the manager, the system begins in state $\mathcal{S}_{00}$. If the worker assigned the task of matching "A" gets one, then the worker passes it to the manager and the state of the system transitions to $\mathcal{S}_{10}$. In the case that during a single simulation time-step both workers get their assigned letter, then the system moves directly to $\mathcal{S}_{11}$.

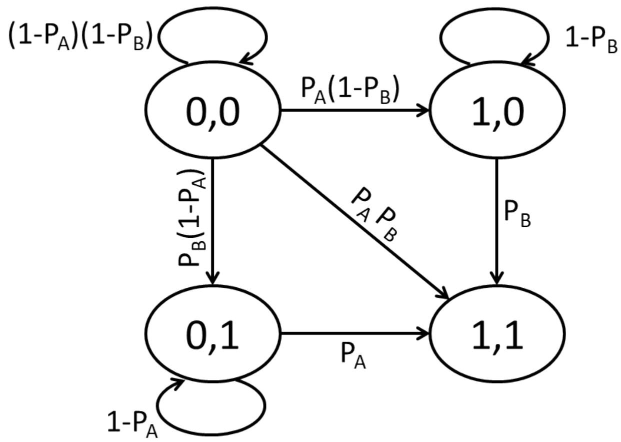


Figure 4.5. Markov Chain of Hierarchy with One Manager and Two Workers

Using Figure 4.5, we derive the analytic formula for $E[N_{00}]$ as follows,

$$\begin{aligned}
E[N_{00}] &= 1 + P_A(1-P_B)E[N_{10}] + P_B(1-P_A)E[N_{01}] + P_A P_B E[N_{11}] \\
&\quad +(1-P_A)(1-P_B)E[N_{00}] \\
&\quad \text{where } E[N_{01}] = 1 + (1-P_A)E[N_{01}] \implies E[N_{01}] = \frac{1}{P_A} \\
&\quad \text{where } E[N_{10}] = 1 + (1-P_B)E[N_{10}] \implies E[N_{10}] = \frac{1}{P_B} \\
E[N_{00}] &= 1 + P_A(1-P_B)\Big(\frac{1}{P_B}\Big) + P_B(1-P_A)\Big(\frac{1}{P_A}\Big) + (1-P_A)(1-P_B)E[N_{00}] \\
&= \frac{(P_A)^2(-P_B) + (P_A)^2 - P_A(P_B)^2 + P_A P_B + (P_B)^2}{P_A P_B(P_A(-P_B) + P_A + P_B)}. \qquad (4.7)
\end{aligned}$$

Using probabilities from the Cornell letter distribution and Equation 4.7 we calculate,

$$\begin{aligned}
E[N_{00}] &= \frac{(P_A)^2(-P_B) + (P_A)^2 - P_A(P_B)^2 + P_A P_B + (P_B)^2}{P_A P_B (P_A(-P_B) + P_A + P_B)} \\
&= \frac{(0.08124)^2(-0.01489) + (0.08124)^2 - 0.08124(0.01489)^2 + 0.0012 + (0.01489)^2}{0.0012(0.08124(-0.01489) + 0.08124 + 0.01489)} \\
&= 68.9332.
\end{aligned}$$

We mathematically calculate the expected number of time-steps needed for the base case hierarchy to complete its objective to be 68.9332.

An alternative way to calculate the expected value is to solve the system of equations defined by the transition submatrix T in Table 4.8.

Table 4.8. Transition Submatrix for One Manager, Two Worker Hierarchy

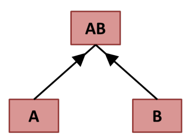
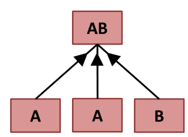
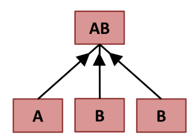
	(0,0)	(0,1)	(1,0)
(0,0)	$(1 - P_a)(1 - P_b)$	$P_b(1 - P_a)$	$P_a(1 - P_b)$
(0,1)		$1 - P_a$	
(1,0)			$1 - P_b$

Solving this system of equations using R, we also obtain $E[N_{00}] = 68.9332$. Both mathematical methods support our word matching game simulation, which resulted in an expected value of 69.3888 ± 1.2861.

If you recall the calculated and simulated results from Section 4.4.2, the base case hierarchy's performance is nearly identical to the performance of an organizational architecture consisting of a single worker with primary task "AB." In the base case hierarchy, the manager's role is simply to collect completed tasks from workers, and therefore the manager receives an "A" or "B" at nearly the same rate as one worker attempting to match "AB" alone. The exception to this statement is the small chance that both workers receive their exact matching letter at the same time (seen as the diagonal line in Figure 4.5). Due to this unlikely but possible transition in the Markov-Chain, our mathematical calculations confirm a very slight advantage favoring the base case hierarchy over the single worker architecture discussed in Section 4.2.2 ($68.9332 < 69.0536$). Unfortunately, the advantage is so small that due to the stochastic nature of the Monte-Carlo simulation, our result of 69.3888 ± 1.2861 has too wide a range to differentiate the two architectures. So why do companies and organizations often have hierarchical architectures? The benefits are easily observable when we consider architectures with three or more workers.

We continue our incremental research approach by adding one more worker to our base case hierarchy. As confirmed in previous sections, additional workers can improve an organization's overall performance. But in a hierarchy, the assignment of tasks to workers, more than the number of workers added, has the greatest impact on an architecture's performance. Table 4.9 compares simulation results of two hierarchies with three workers with the base case hierarchy. Both larger hierarchies show improved performance, but the task assignment ("A" or "B") of the newly added worker makes a tremendous difference.

Table 4.9. Simulation Results of One Way Hierarchies with Task = "AB"

Architecture:			
Expected Value:	69.3888 ± 1.2861	67.1308 ± 1.3182	36.6723 ± 0.6139

The assignment of worker tasks in accomplishing a specific objective has significant impact on performance. As Table 4.9 shows, if the organization's objective is "AB" then having two workers assigned to match "B" vastly outperforms an architecture with two workers assigned "A." In terms of our word matching simulation, knowing the particulars of the Cornell letter distribution allows a keen manager to assign tasks to his/her workers in the best manner. Since the probability of getting a "B" (0.01489) is smaller than getting an "A" (0.08124), it makes sense to assign more workers the more difficult task. If an organization's operating environment (letter distribution) is known, optimum worker task assignment is possible. If more workers are added to an architecture, the operating environment changes, or the organization's objective changes, task assignments should be reevaluated or the organization's performance may suffer. When organizational architectures are designed (and worker tasks assigned) around completing a specific objective while operating in a known environment, performance can be optimized, but the organization's robustness to changing objectives or environments decreases. Next, we explore hierarchies with a more advanced cooperation rule that increases performance and robustness.

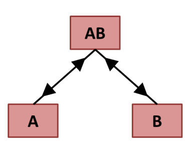
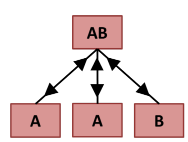
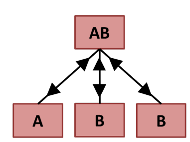
4.5.2 Base Case Modification - Two Way Vertical Communication

Next, we modify the base case hierarchy to allow the manager to share letters received from one worker with other workers. For example, using our base case hierarchy with objective "AB," if the worker assigned to match "A" receives a "B" then that worker passes it up to the manager who then passes it to the worker with task "B" on the next simulation time step.

We first measure the performance of a two worker, one manager hierarchy in which the manager can share letters. Our simulation yields an expected value of 34.9463 ± 0.6484 which is nearly identical to the simulation results in Section 4.4.2 of 34.8019 ± 0.6404 for two fully connected workers. The two architectures perform nearly identically except for the slight advantage of the hierarchy in the case of both workers getting their exact matching letter at the same time. Again, the advantage is so small that it falls within the simulation's 95% confidence interval half width.

Table 4.10 shows simulation results for a two worker hierarchy and two versions of a three worker hierarchy we explore next. In Section 4.5.1, when a third worker is added to the base case hierarchy, performance improves but it varies based on which task the new worker was assigned. We might expect the same result again but our simulation results show that assigning either task to the new worker improves the organization's performance nearly equally. The reason for this is the manager can share letters received by workers who don't need them with a worker who does. Therefore, the three workers perform as if they are fully-connected, but with fewer connections. Keeping the number of connections in an architecture to a minimum is important in order to reduce unnecessary costs associated with cooperation and communication.

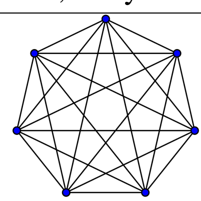
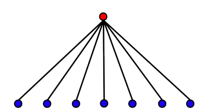
Table 4.10. Simulation Results of Two Way Hierarchies with Task = "AB"

Architecture:			
Expected Value:	34.9463 ± 0.6484	23.5521 ± 0.4289	23.6020 ± 0.4365

As introduced in Section 4.4.2, connections between agents in an architecture allow communication and cooperation to occur but have an associated cost. In the smaller architectures we examine, the difference in number of connections between hierarchical and fully connected architectures is minor, but in larger organizations the difference is significant.

Consider the two architectures (each with k workers) shown in Table 4.11. In the first architecture, each of the k workers is connected to all the others. In the second, each of the k workers is connected to a single manager. The first architecture, with k workers connected to all the others has $k(k-1)/2$ connections. The second architecture is a hierarchy, consisting of one manager connected to k workers, and has only k connections.

Table 4.11. Comparison of Number of Connections Between Seven Worker Hierarchy and Seven Fully-Connected Workers

	k Workers, Fully-Connected	Hierarchy with k Workers
Architecture:		
# of Connections:	$k(k-1)/2$	k
$k = 7$	21	7

At low values of k, the difference in number of connections is small. But in larger organizations, architectures can grow dense with connections quickly, thus negatively impacting performance. Figure 4.6 demonstrates the exponential versus linear growth in the number of connections between fully-connected and hierarchical architectures.

As the size of any organization increases, the number of connections will certainly increase. Limiting extraneous connections will reduce costs associated with communication and cooperation, thereby improving overall organizational performance. As we continue this research, we examine the careful balance between reducing connections in order to increase performance, with the impact on an architecture's robustness to change.

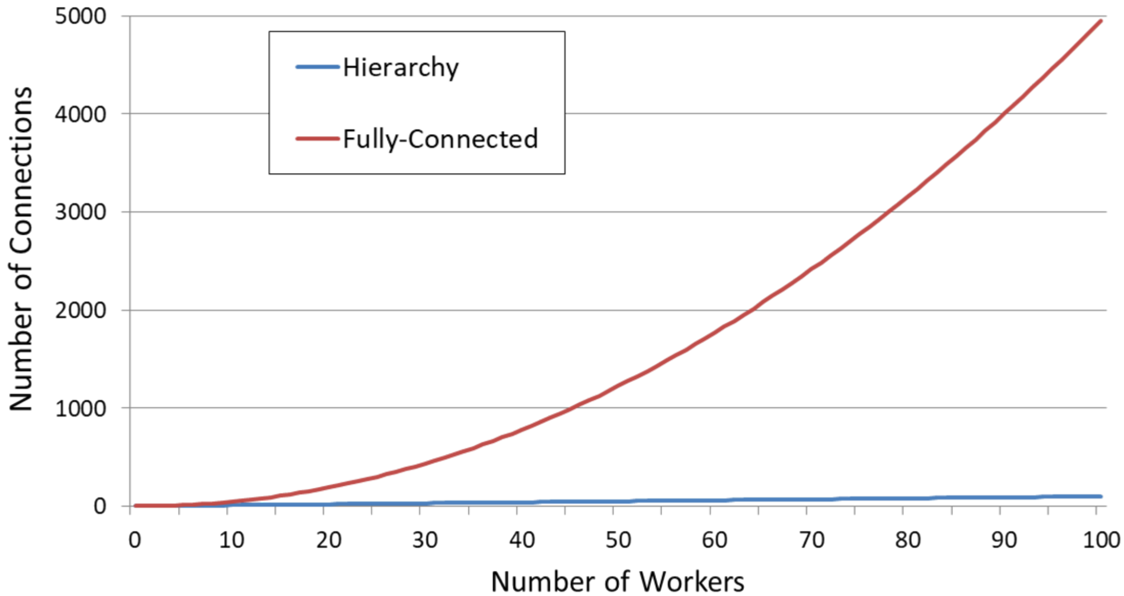
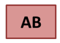
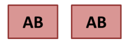
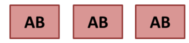
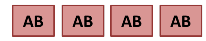
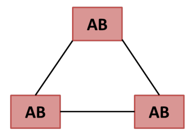
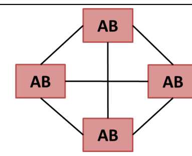
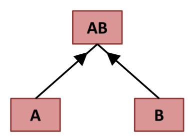
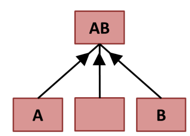
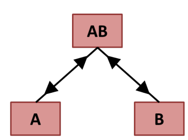
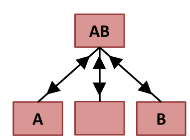


Figure 4.6. Graph Showing an Exponential Increase in Number of Connections in a Fully-Connected Architecture Vs. a Linear Increase in a Hierarchy

We present Table 4.12 to summarize the architectures we have examined. The bracketed letter following each result annotates the method used to calculate the expected value.
A = Analytic equations and S = Simulation.

Table 4.12. Architecture Performance Summary

	1 Agent	**2 Agents**	**3 Agents**	**4 Agents**
Independent Agents	 69.0536 [A] 68.7626 ± 1.2761 [S] We begin with the simplest architecture consisting of one agent.	 36.4587 [A] 36.6198 ± 0.6240 [S] Doubling the number of agents reduces the expected value nearly in half.	 26.3074 [A] 25.8209 ± 0.4025 [S] More agents yield better performance, but benefit begins to taper off.	 20.9992 [A] 20.9587 ± 0.3104 [S] Additional agents continue to improve performance, but it does so less and less.
Fully Connected Agents		 34.7854 [A] 34.8019 ± 0.6404 [S] Connecting two agents allows them to share letters, thus improving performance compared to two independent agents.	 23.2637 ± 0.4259 [S] Connecting three agents continues to decrease the expected value. Realistically, now sharing rules are needed to avoid creating copies of letters. Time deciding how to share, is time an agent is not matching new letters.	 17.7431 ± 0.3175 [S] Using simple sharing rules, additional agents result in improved performance but the rate of improvement slows.
Hierarchy One-way Comms			 69.3888 ± 1.2861 [S] With one-way communication, performance is nearly identical as a single agent architecture.	 A = 67.1308 ± 1.3182 [S] B = 36.6723 ± 0.6139 [S] Assigning "A" or "B" to the added worker makes a huge difference. When the environment is known, tasks can be assigned to optimize performance.
Hierarchy Two-way Comms			 34.9463 ± 0.6484[S] The expected value is nearly same as two connected agents since the manager can share letters with other agents.	 A = 23.5521 ± 0.4289 [S] B = 23.6020 ± 0.4365 [S] With two-way communication, the architecture's sensitivity to task assignments is much less, thus making the architecture more robust.

4.5.3 Challenges Moving Forward

As introduced in Section 4.4.2, communication and cooperation between agents has associated costs to an organization's overall performance. We model communication and cooperation costs by giving agents a finite work capacity on each time step of the simulation. In other words, we attempt to model real-life by limiting every worker and manager to a maximum amount of work he or she is able to complete in a given time period. To justify this concept, consider a simple hierarchy in which a single manager has infinite work capacity. If this were the case, because the manager is connected to every worker and has no work limit, the manager can share as many letters that are available across the architecture, effectively performing equally to a fully-connected architecture. If this were the case in real life, hierarchies would never need to grow beyond two levels (one manager connected to k workers), which we know is not true.

Because we model communication and cooperation costs using agent work capacities, if the number of workers assigned to a single manager grows and grows, the manager will eventually become overwhelmed. When the quantity of letters needing to be shared (received from workers) exceeds the manager's work capacity, sharing gets delayed to subsequent simulation time steps. This insight means there exists a threshold at which adding more workers to a single manager stops improving system performance.

We claim that workers and managers have a limitation of how much work they can accomplish in a given time period. In terms of the word matching game simulation, when an agent's capacity is exceeded, new letters from the distribution are missed. We could run simulations to explore this claim and determine the optimum number of workers to connect to a manager, but that number would only apply to that specific architecture, set of agent work capacities, organizational objective, and operating environment. Moreover, real organizations rarely have the problem of having too many employees. More likely, organizations strive to find the best set of tasks to assign workers and number of worker connections that optimizes the organization's performance. This is what we will explore in Section 4.6.

Challenges of examining more complicated architectures.

Previously in our research we analyzed architectures using one or both of the following methods: 1) analytic mathematical equations, or 2) computer simulation using the word matching game. When possible we solved mathematical equations directly, and for more complicated systems of equations, we relied on a computer-aided matrix solver. Unfortunately, as discussed in Section 4.3.2, as the number of agents in an architecture or the length of an organization's objective increases, the mathematical equations and/or submatrix T necessary to represent more complicated organizations quickly grow in size and complexity. In previous sections, we examine trivial and base case architectures that allow tractable use of multiple analysis methods. An intentional by-product of doing so is to provide credibility to the computer simulation we developed. We admit simulation results not supported with analytic calculations are less convincing, but they remain informative when analytic approaches pass a reasonable level of complication and scale.

4.6 Exploring Architecture Robustness

In this section, we consider the issue of organizational robustness, namely how an architecture performs in the presence of either (1) changing objectives or (2) changing environments. An architecture is considered robust if it can perform nearly as well given any objective or operating environment. In terms of our game, a very robust architecture would be able to effectively match any word given any letter distribution.

We begin this discussion by presenting a series of organizational architectures, in which one manager is connected to as many as six workers in a two-tier hierarchy. We begin with a simple three worker architecture as shown in Figure 4.7.

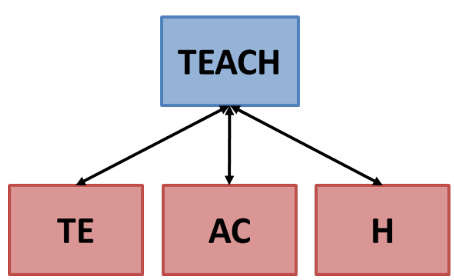


Figure 4.7. Two-Tier Hierarchy with One Manager and Three Worker Agents

In terms of our word matching game, the objective of this architecture is to match the word "TEACH." Workers are assigned a primary task of one or two letters to match. Once matched, they pass completed tasks to the manager. Workers are also aware of the overall objective so they have a secondary task to pass to the manager any letters needed by other workers. When the manager receives such a letter, it is given on the next turn to the worker who needs it most, thus replacing the random letter the worker would have received from the distribution on that turn.

Again, let $\mathcal{S}$ represent the state of space of the system. From the manager's point of view, the system begins in state $\mathcal{S}_{000}$, meaning the manager has no matches for "TE," "AC," or "H." Using the word matching game and the Cornell letter distribution, we simulated $E[N_{000}] = 23.3745 \pm 0.3071$. In other words, on average the architecture takes approximately 23 discrete time steps to match "TEACH." Without an analytic method of verifying the simulation's output due to the complex mathematics, we will not attempt to prove its accuracy, instead we use it as benchmark for comparison in subsequent experiments.

Now, imagine the organization hired three additional workers, one for each of the three primary tasks, as shown in Figure 4.8.

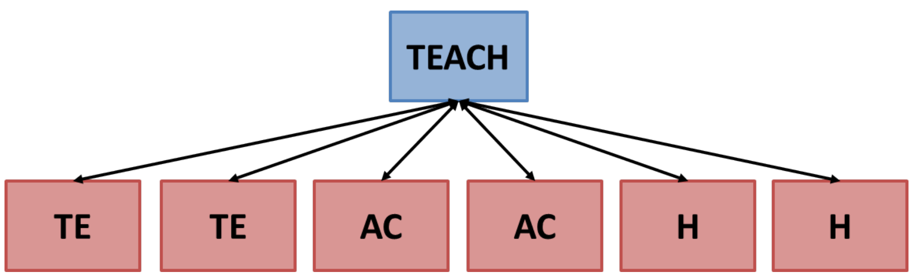


Figure 4.8. Two-Tier Hierarchy with 1 Manager and 6 Worker Agents

Effects of Changing Task Assignments. The selection of task assignments has a direct impact on an architecture's performance. By varying the worker task assignments depicted in Figure 4.8, we use the word matching game simulation to show how an architecture's performance can change, even when the objective and environment remain the same.

Table 4.13 shows that of the four combinations of task assignments, the set of tasks on the bottom row [TE,TE,TE,TE,AC,H] is best. This is evidence of how an organization can customize worker task assignments in its architecture when the operating environment is known in order maximize performance. By assigning more workers to complete *difficult* tasks, the organization's performance increases. However, such specialized organizations face higher risks of performance degradation due to their reduced robustness to change.

Table 4.13. Simulation Results When Varying Worker Tasks Using the Cornell Distribution

Worker Number and Task						
1	2	3	4	5	6	Expected Completion Time
TE	AC	AC	H	H	H	12.1283 ± 0.1564
TE	TE	AC	AC	H	H	11.4811 ± 0.1428
TE	TE	TE	AC	H	H	11.2224 ± 0.1394
TE	TE	TE	TE	AC	H	10.5901 ± 0.1259

The best set of task assignments depends on how dynamic an organization's environment and objective are and how willing an organization is to accept performance degradation. If the environment or objective can change fluidly, then a specialized architecture with less robustness is a poor choice. If an organization knows (or controls) its environment, it can design an optimal architecture and assign worker tasks optimally.

Effects of a Changing Environment. Specialization of an architecture to a specific organizational objective and/or operating environment can increase efficiency, but at the expense of being less robust. Let us now examine the "TEACH" architecture using a new letter distribution. In this distribution, named the *Even Distribution*, each of the 26 letters in the alphabet has an equal probability of being chosen ($1/26 = 0.03846$). Using the Even Distribution and the word matching game simulation, Table 4.14 shows the results of the same four experiments listed in Table 4.13, but using the new letter distribution to simulate a change in the architecture's operating environment.

Table 4.14. Simulation Results When Varying Worker Tasks Using the Even Distribution

Worker Number and Task

1	2	3	4	5	6	Expected Completion Time
TE	AC	AC	H	H	H	12.1883 ± 0.1069
TE	TE	AC	AC	H	H	12.0773 ± 0.1072
TE	TE	TE	AC	H	H	12.4031 ± 0.1090
TE	TE	TE	TE	AC	H	12.8733 ± 0.1101

The simulation results in Table 4.14 show a decrease in performance compared to the same sets of task assignments measured using the Cornell distribution. By comparing the bottom row of each table [TE,TE,TE,TE,AC,H] we see an approximately 22% decrease in performance when we change the environment by running the simulation with the Even distribution. Next, imagine a hypothetical letter distribution in which T, E, A, C, and H happen to be the five rarest letters. Without simulating this scenario, we know this distribution would result in a significant decrease in the "TEACH" architecture's performance.

Note that the simulated results in Table 4.14, are all very close and have less variability than in Table 4.13. Because no letter in the Even distribution is more or less probable than another, the best architecture is one that divides tasks evenly among workers. For this reason, the best choice of worker assignments for the Even distribution is [TE,TE,AC,AC,H,H], which our simulation results support.

When an organization operates in an environment prone to change, it must adapt an architecture that is not too specialized. By assigning tasks evenly among workers, an organization's performance may not be the highest possible, but it is more robust in performing in a variety of possible operating environments.

Effects of a Changing Objective. If an organization's objective is subject to change, a specialized architecture will suffer a decrease in performance if the objective changes. Consider an automobile factory that produces cars. How would you expect the factory to perform the next day if it was required to begin making trucks? If not allowed to stop manufacturing to reorganize tasks and processes (change its architecture), the factory's output would greatly suffer. Now, imagine that instead of changing to building trucks, the factory was forced to make boats. The factory's output would likely be lower or even cease all together.

In terms of our word matching game, let us explore how worker task assignments impact an architecture's robustness to a changing objective. First, to be robust, an architecture must not have tasks that are too specific thus preventing an architecture from completing a new objective. For example, consider the difficulty the "TEACH" architecture (Figure 4.8) would have if the organization's objective changed to "CATCH." The new objective contains common letters with the original, but without changing the set of assigned tasks, the existing architecture cannot match "CATCH." This is because TE, AC, and H are too specific of tasks to be capable of matching "CATCH." Even the word "EACH" cannot be matched using the architecture shown in Figure 4.8.

For an architecture to avoid significant performance degradation when its objective changes, its set of assigned tasks must not be too narrow or specific. Let [T, E, A, C, H] be the new set of available tasks that can be assigned to workers. Doing so requires a minimum of five workers in the architecture, but now "TEACH," "CATCH," and "EACH" are objectives that can be matched, thus increasing robustness. Table 4.15 shows the results of using the word matching game to measure the architecture's performance in matching three similar but different objective words. The simulation uses the Cornell letter distribution and the set of tasks [T, E, A, C, H, H], assigned to workers one through six, respectively.

Table 4.15. Simulation Results of Architecture with Task Set [T,E,A,C,H,H]

Objective:	TEACH	CATCH	EACH
Expected Value:	9.6290 ± 0.1248	12.0540 ± 0.1518	14.4724 ± 0.1980

Obviously there are countless more experiments pertinent to exploring and measuring organizational architectures, however the next step one might take is to explore the impact of assigning one or more workers in the "TEACH" architecture as a non-assigned, flexible, *floater* agent. A floater agent, in terms of our word matching game, can match any letter needed that is not already assigned as a task to another worker. Figure 4.9 illustrates such an architecture that can match its primary objective ("TEACH"), but also has the flexibility to accomplish other objectives such as "BEACH," "TEACHER," and even "ZEBRA," therefore increasing the architecture's robustness.

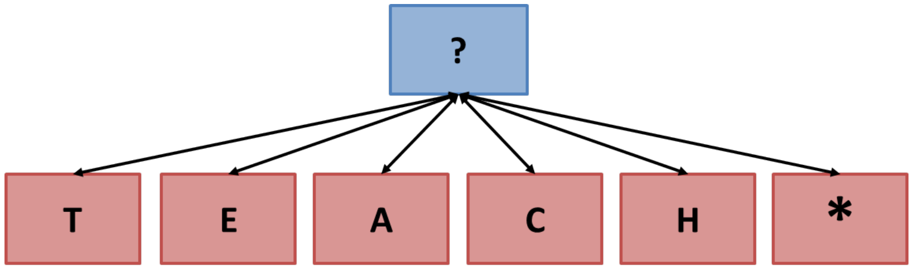


Figure 4.9. Two-Tier Hierarchy with One Manager, Five Worker Agents, and One Floater

With even one floater agent, this architecture can match any objective. If a new objective happens to align well with the architecture's existing worker assignments (like "BEACH" or "TEACHER") then the floater must only match a single extra letter. If the word is poorly suited for the current task assignments such as the word "ZEBRA," then the floater will have to independently match all three new letters. In this case, the organization will eventually complete the objective, but performance would suffer compared to objective words with more letters common to those that are already assigned to workers.

In summary, we assert that organizations with a customized architecture tailored to complete a specific objective in a stable environment will perform most efficiently, however performance will suffer in the case of volatility in the objective or environment. Organizations willing to accept lower or zero performance in order to reorganize its architecture to adapt to a new objective or environment will benefit from specialized hierarchies, known for high efficiency, but low robustness. On the other hand, organizations that must accomplish varying objectives or that operate in dynamic environments must choose a less specialized set of worker tasks.

There exists an unlimited number of specific architectures in use by organizations today. Arguably, all are orders of magnitude more complicated than any architecture we've examined in our research, however they are comprised of the building blocks and fundamental relationships we present. While some organizations specialize at performing a specific task over and over (e.g., manufacturing), others survive by adapting and innovating new products to satisfy ever-changing consumer desires (e.g., technology innovators). The architecture type that is best for any organization depends on how robust it must be to mitigate the risk of poor performance in face of changes or how quickly and correctly the organization can reorganize to adapt to changes in its objective and/or environment. In the final chapter, we revisit the features and traits of such architectures, like those that General McChrystal implemented to succeed in Iraq.

CHAPTER 5:
Conclusion

5.1 Conclusion

This thesis presents a quantitative framework, in the form of a game, that imitates the basic processes of an organization. The game is supported by mathematical analysis and measures how well an organizational architecture performs in completing its objective in a particular environment. We explore how to quantitatively measure an architecture by examining simple architectures and incrementally expanding our research. By doing so, we observe how different architectures react to change and what features of an architecture most impact its robustness to change.

First, we show that adding independent (non-connected) workers to an architecture improves overall performance, but at a much slower rate than if the same workers were connected in a manner that allows them to communicate and cooperate. However, challenges arise as an architecture grows in size. Adding workers can result in exponential growth in the number of connections which eventually limits the performance of an architecture, despite the number of workers added. This is because workers have a finite capacity of work, and effort spent cooperating or communicating consumes some of that capacity. Eventually, once the size of a fully-connected architecture grows beyond a threshold, performance increases are not possible. To avoid this, larger architectures take advantage of carefully selecting connections between workers. These architectures maximize the performance advantages of connections while minimizing costs.

A hierarchy is one such architecture with minimal connections. We use the word matching game simulation to test hierarchies with varying worker assignments and contrast performance results. When splitting up an objective into smaller tasks to assign to workers, we show how sensitive an architecture's performance is to the environment and objective. If either change, performance suffers. Such architectures are less robust to change when they are customized to a specific objective and/or environment. With this foundation and framework established, we gain insight into why the JSOTF originally failed in Iraq and why General McChrystal's "team of teams" concept was a better fit.

Like a customized hierarchy tailored toward a specific objective in a stable environment, the JSOTF's performance against AQI was originally unsuccessful because its architecture was too specialized to perform specific tasks which made it less robust to external changes. Its architecture was defined by a centralized structure with a strict hierarchy of authority. The task force was comprised of specialized units that didn't have the connections and processes to share information or cooperate effectively due to the vertical nature of its architecture. This research addresses why, while being extremely efficient at their assigned objectives, organizational architectures like that of the JSOTF, are less effective against an unpredictable and agile enemy.

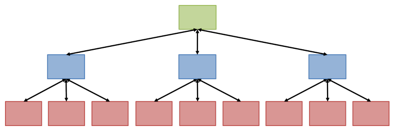
As General McChrystal describes in *Team of Teams*, his restructuring efforts changed the JSOTF's organizational architecture. The architecture became decentralized with decision making authorized at lower levels, thereby freeing up available work capacity for senior leaders. Additionally, the task force's architecture maximized the advantages of cooperation while limiting connections between workers by using teams and connecting them horizontally with liaison officers. McChrystal's new task force saw increased effectiveness fighting a fluid enemy in multiple arenas. Although its architecture was potentially less efficient at performing certain specific tasks than before, the collaborative and cooperative characteristics of the new architecture became more robust and therefore more effective in defeating a dynamic enemy like AQI.

5.2 Future Work

Options for future work include measuring the performance of more sophisticated architectures. Such architectures might include the following.

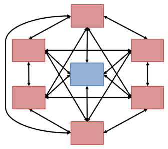
Multi-Tier Hierarchy

We recommend building upon the two-tier hierarchy to study architectures with multiple tiers. Of particular interest would be a sensitivity analysis of how many workers and/or managers should occupy each tier to maximize performance.



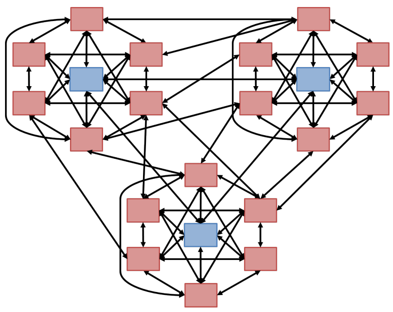
Team

We recommend the use of simulation to explore architecture design elements for more organic and collaborative architectures. We recommend testing to see whether or not a threshold exists at which performance gains cease as the size of a team architecture grows.



Team of Teams

A significant goal for future research would be to connect multiple teams and measure the robustness of new architectures of this type to changing objectives and/or environments.



To implement these or other more complicated architectures in the future, the word matching game simulation would require enhancement. One addition might include adding a new type of agent, such as a *Leader*, who is superior to *Manager* agents and has increased work capacity. Other embellishments might consider changes to facilitate longer length, multi-word objectives.

5.3 Final Thoughts

One can imagine numerous additional experiments to better understand and identify features that compose a *best* (most robust) architecture for accomplishing varying objectives in dynamic environments. However, there is growing evidence (Eisenberg et al., 2014) to suggest that the best architecture needs not be robust, but resilient. Contemporary discussions concerning resilient organizations are less about finding the most robust architecture, but understanding when to change an architecture's features. Eisenberg et al. (2014) asserts that resilience depends upon an architecture's flexibility and interconnectedness. A resilient architecture doesn't have to mitigate all risk, but its features must be able to change fast enough and correctly, for the organization to quickly recover after unforeseen challenges.

In discussing resiliency, Park et al. (2013) introduces four recursive processes that can lead to the emergence of resilience in organizations.

1. **Sensing.** Observing the operational environment for changes and incorporating observable information into the current understanding. "This process connects components in the physical domain to the information domain" (Eisenberg et al., 2014).
2. **Anticipation**. Imagining future possible states. "This process connects components in the information domain to the cognitive domain" (Eisenberg et al., 2014).
3. **Adaptation.** Changing as a result of sensing or anticipating. "This process connects the cognitive domain to the physical domain" (Eisenberg et al., 2014).
4. **Learning.** Creating new knowledge or improving an organization's understanding based on past actions. "This process connects the physical, information, and cognitive domains together" (Eisenberg et al., 2014).

The concepts and experiments in this research help orient future discussion of C2 architectures with an operations research perspective. Our framework allows quantitative analysis to compare and contrast architectures. By doing so we can more precisely discuss which architecture components and processes best serve an organization's goals.

Although motivated by McChrystal's story, the most important piece of the *Team of Teams* narrative is not the final topology of the task force's architecture, but rather the importance of an organization's ability to identify the need for change, know what protocols are necessary to effect the needed change, and the ability to initiate the change in time for it to be relevant.

APPENDIX A:
Lexicon

Agent. Each *Organizational Architecture* measured is comprised of one or more Agents. In this research, Agents have two possible types: (1) Worker or (2) Manager. Every Agent has a Primary Task, Secondary Task, and Work Capacity. Relationships and connections between Agents in an *Organizational Architecture* are defined using three lists: (1) Parents, (2) Children, and (3) Peers.

Agent Tasks. An Agent's Primary Task is the one or more letter component of the *Objective* that the *Organizational Architecture* is trying to match. Secondary Tasks are defined as required, to simulate other rules or processes of an architecture (e.g., sharing and filtering).

Architecture Robustness. How well an *Organizational Architecture* can complete its *Objective* in the presence of either (1) changing *Objectives* or (2) changing *Environments*. (See *C2 Effectiveness*.) An architecture is considered robust if it can perform nearly as well given any *Objective* or *Environment*.

Command and Control (C2). "Means by which a commander synchronizes and/or integrates for activities in order to achieve unity of command" (United States Joint Chiefs of Staff, 2017). Painter et al. (2009) describes C2 using five domains: (1) Structure, (2) Environment, (3) Work Processes, (4) Human Resources, and (5) Culture.

C2 Architecture. See *Organizational Architecture*.

C2 Effectiveness. "The net result of the successful interaction of a complex architecture that is comprised of people, procedures, and equipment" (Bethmann and Malloy, 1989).

Communication and Cooperation. In this research, Communication and Cooperation represent any type of action an *Agent* would do that involves interaction with another *Agent*. Types of such interactions are commonly described as Vertical or Horizontal.

Environment. "Everything that exists outside the boundary of the organization and has the potential to affect all or part of the organization" (Daft, 2001). See *Game Environment*.

Fully-Connected Architecture. An architecture type characterized by two or more *Agents* that are connected to all other *Agents*. These architectures are dense with connections and have mostly Horizontal *Communication and Cooperation* rules.

Game. A Game, defined by Fullerton (2014), contains five major elements: (1) Components, (2) Space, (3) Goals, (4) Mechanics, and (5) Rules. This research presents a Word Matching Game used to measure the performance of an *Organizational Architecture*. It measures how many simulation time-steps it takes to complete an *Objective* in a specific *Environment*.

Game Objective. A word or combination of letters the *Organizational Architecture* being measured must match. It is comprised of one or more *Agent Tasks*.

Hierarchical Architecture. An architecture type characterized by one of more *Agents* connected to a single *Agent* using minimal connections and mostly Vertical *Communication and Cooperation* rules.

Objective. The mission or goal an organization is designed to achieve. See *Game Objective*.

Organization. "(1) Social entities that (2) are goal oriented, (3) are designed as deliberately structured and coordinated activity systems, and (4) are linked to the external environment" (Daft, 2001).

Organizational Architecture (C2 Architecture or Organizational Structure). An organization's architecture or structure is commonly represented on a chart that uses boxes and lines to group organization members into assigned roles, depict formal reporting relationships, and state spans of control. Additionally, "organizational architecture is about the structure, processes, organizational roles, power and authority, reporting relationships" (Galbraith et al., 2001). Alberts and Hayes (2006) describes three dimensions of Organizational Architecture: "Allocation of Decision Rights," "Distribution of Information," and "Patterns and Policies Governing Interactions."

APPENDIX B:
Cornell Letter Distribution

The Department of Mathematics of Cornell University in 2004 published the following list of letters and their frequency of occurrences in a 40,000-word English dictionary.

Table B.1. Cornell University Letter Distribution

Letter	Frequency	Probability	Letter	Frequency	Probability
A	14810	0.081238378	N	12666	0.069477738
B	2715	0.014892788	O	14003	0.076811682
C	4943	0.0271142	P	3316	0.018189498
D	7874	0.043191829	Q	205	0.001124502
E	21912	0.120195499	R	10977	0.060212942
F	4200	0.023038568	S	11450	0.062807524
G	3693	0.020257483	T	16587	0.090985886
H	10795	0.059214604	U	5246	0.028776268
I	13318	0.073054201	V	2019	0.011074969
J	188	0.00103125	W	3819	0.02094864
K	1257	0.006895114	X	315	0.001727893
L	7253	0.039785412	Y	3853	0.021135143
M	4761	0.026115862	Z	128	0.000702128

The *Probability* column is calculated by dividing each letter's frequency by the total sum of both *Frequency* columns (182,303).

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Alberts D, Hayes R (2003) *Power to the Edge: Command and Control in the Information Age* (Assistant Secretary of Defense (Command Control Research Program), Washington D.C.).

Alberts D, Hayes R (2006) *Understanding Command and Control* (Assistant Secretary of Defense (Command Control Research Program), Washington D.C.).

Alderson D (2008) Catching the "network science" bug: Insight and opportunity for the operations researcher. Operations Research, INFORMS, Catonsville, MD, 56(5):1047–1065.

Arquilla J, Ronfeldt D (2001) *Networks and Netwars: The Future of Terror, Crime, and Militancy* (Rand Corporation, Santa Monica, CA).

Ashworth M, Carley K (2007) Can tools help unify organization theory? Perspectives on the state of computational modeling. Computational & Mathematical Organization Theory, Springer, New York, 13(1):89–111, https://link.springer.com/article/10.1007/s10588-006-9000-9.

Barabási A (2016) *Network Science* (Cambridge University Press, Cambridge, United Kingdom).

Bethmann R, Malloy K (1989) Command and control: An introduction. Master's thesis, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/26924.

Breton R (2008) The modeling of three levels of cognitive controls with the cognitive-OODA loop framework. DRDC Valcartier Technical Report 2018-111, Defence R&D Canada-Valcartier, Ontario, Canada, http://cradpdf.drdc-rddc.gc.ca/PDFS/unc95/p530447_A1b.pdf.

Breton R, Rousseau R (2006) Modeling approach for team decision-making. DRDC Valcartier Technical Report 2003-368, Defence R&D Canada-Valcartier, Ontario, Canada, http://cradpdf.drdc-rddc.gc.ca/PDFS/unc46/p525056.pdf.

Carey A (1967) The Hawthorne Studies: A Radical Criticism. American Sociological Review, JSTOR, New York, 32(3):403–416.

Carley K, Wallace W (2001) Computational organization theory. Encyclopedia of Operations Research and Management Science, Kluwer Academic Publishers, Norwell, MA, 1(1):126–132.

Cornell Department of Mathematics (2004) English letter frequency (based on a sample of 40,000 words), https://www.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html.

Daft R (2001) *Essentials of Organization Theory and Design* (South Western Educational Publishing, Mason, OH).

Daft R (2012) *Organization Theory and Design* (Nelson Education, Scarborough, Ontario, Canada).

Eisenberg D, Linkov I, Park J, Seager T (2014) Resilience metrics: Lessons from military doctrines. Solutions, Shelton, CT, 5(5):76–87, https://www.thesolutionsjournal.com/article/resilience-metrics-lessons-from-military-doctrines.

French W, Bell C (1999) *Organization Development: Behavioral Science Interventions for Organizational Improvement* (Prentice-Hall, Upper Saddle River, NJ).

Fullerton T (2014) *Game Design Workshop: A Playcentric Approach to Creating Innovative Games* (CRC Press, Boca Raton, FL).

Galbraith J, Downey D, Kates A (2001) *Designing Dynamic Organizations: A Hands-On Guide for Leaders at All Levels* (HarperCollins Leadership, Nashville, TN).

Lederman G (1999) *Reorganizing the Joint Chiefs of Staff: The Goldwater-Nichols Act of 1986* (Greenwood Publishing Group, Westport, CT).

Lee J, Carley K (2004) *OrgAhead: A Computational Model of Organizational Learning and Decision Making* (Carnegie Mellon University, Pittsburgh, PA).

Lin Z, Carley K (1995) DYCORP: A computational framework for examining organizational performance under dynamic conditions. Journal of Mathematical Sociology, Taylor & Francis, Milton Park, United Kingdom, 20(2-3):193–217.

McChrystal S, Collins T, Fussell C, Silverman D (2015) *Team of Teams: New Rules of Engagement for a Complex World* (Penguin, London, United Kingdom).

Mintzberg H (1989) The structuring of organizations. Strategic Management, Springer, New York, 1:322–352.

Morgan G (2006) *Images of Organization* (Sage Publications, Thousand Oaks, CA).

Nadler D, Gerstein M, Shaw R (1992) *Organizational Architecture: Designs for Changing Organizations* (Jossey-Bass Publications, Indianapolis, IN).

Painter D, Weaver M, White S (2009) Reorganizing for irregular warfare, Master's thesis, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/4353.

Park J, Seager T, Rao P, Convertino M, Linkov I (2013) Integrating risk and resilience approaches to catastrophe management in engineering systems. Risk Analysis, Society for Risk Analysis, McLean, VA, 33(3):356–367.

R Core Team (2016) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org.

Roethlisberger F, Dickson W (1939) *Management and the Worker* (Harvard University Press, Cambridge, MA).

Rossum GV, Drake F (2003) *Python Language Reference Manual* (Network Theory Ltd, Godalming, United Kingdom).

Taylor F (1911) *The Principles of Scientific Management* (Harper Brothers, New York).

United States Joint Chiefs of Staff (2017) *Doctrine for the Armed Forces of the United States, Joint Publication 1* (Joint Chiefs of Staff, Washington, DC).

Weber M (1922) *Economy and Society* (University of California Press, Oakland, CA).

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California